

Diseño e implementación de un prototipo para el monitoreo y control de temperatura mediante
el sistema de secado al sol para el café

Presentado por

Hector Fabio Duran Montes

Universidad Nacional Abierta y a Distancia “UNAD”

Escuela de Ciencias Básicas, Tecnologías e Ingenierías “ECBTI”

Abril del 2019

Pitalito - Huila

Diseño e implementación de un prototipo para el monitoreo y control de temperatura mediante
el sistema de secado al sol para el café

Presentado por

Hector Fabio Duran Montes

Trabajo de opción de grado presentado como requisito para optar al título de

Ingeniero de Sistemas

Director

Diego Fernando Nava Cuevas

Universidad Nacional Abierta y a Distancia “UNAD”

Escuela de Ciencias Básicas, Tecnologías e Ingenierías “ECBTI”

Abril del 2019

Pitalito - Huila

Aceptación

Dedicatoria

Dedico este proyecto a mi familia en especial a mis padres, el cual han sido el pilar principal de mi formación personal y profesional. Gracias por brindarme su apoyo incondicional en todo momento a pesar de las circunstancias y adversidades que se han presentado en este largo trayecto académico.

A mi hermano y mi novia por su comprensión, sus consejos para mejorar en cada proceso afrontado que sin esperar nada a cambio siempre estuvieron ahí para brindarme su apoyo y su colaboración en todo momento.

Le agradezco a mi institución y mis maestros por brindarme su ayuda y esfuerzo por compartir todo su conocimiento para formarme como un excelente profesional y para poder culminar exitosamente mi profesión.

Agradecimientos

Agradezco primero que todo a Dios por brindarme salud, conocimiento y amor sobre todas las cosas que me rodean. Por guiarme a lograr exitosamente cada meta propuesta a corto y largo plazo. También agradezco a mis compañeros el cual fueron parte fundamental en este proceso académico en aquellas dificultades y fortalezas para cumplir este sueño de ser profesional.

Tabla de contenido

Introducción	13
Capítulos	14
Capítulo 1. Aspectos generales	14
1.1 Selección y definición del tema de investigación	14
1.2 El problema de investigación	15
1.3 Línea de investigación	18
1.4. Objetivos de la investigación	18
1.5 Justificación de la investigación	20
1.6 Aspectos Metodológicos	25
Capítulo 2. Marco de referencia de la investigación.....	29
2.1 Marco teórico.	29
2.2 Marco conceptual.....	56
Capítulo 3. Análisis del sistema actual	62
3.1 Identificación de las necesidades	62
3.2 Análisis Económico y Técnico	62
3.3 Arquitectura de sistemas (Diagrama de Flujo de Datos – Diagrama de Contexto del Sistema Actual)	69
3.4 Especificación de la arquitectura de sistemas (Explosión del DFD - Diccionario de Datos)	71
Capítulo 4. Requerimientos del Sistema Propuesto	75
4.1 Creación de prototipos	75
4.2 Requerimientos de entrada/salida	75
4.3 Requisitos de hardware/software	77
Capítulo 5. Diseño de sistemas	79
5.1 Diseño de salida	79
5.2 Diseño de entrada.....	80
5.3 Diseño de base de datos (Modelo Entidad/Relación)	81
5.4 Diseño de controles	82
5.5 Diseño de procedimientos	85
Capítulo 6. Desarrollo e implementación del sistema	86
Puesta en marcha.....	86
Desarrollo del prototipo	86

Resultados	147
Recomendaciones	149
Conclusiones	150
Referencias Bibliográficas	152
Anexo 1 - Encuesta	157
Anexo 2 – Resultados Encuesta	159

Tablas

Tabla 1 Síntomas, Causas, Pronóstico y Control pronostico	16
Tabla 2 Enfermedades causadas por las altas temperaturas.....	21
Tabla 3 Características específicas de las diferentes placas Raspberry pi.....	32
Tabla 4 Características Técnicas del módulo ESP8266.....	33
Tabla 5 Características del Sensor DHT11	35
Tabla 6 Características del Sensor DHT22.....	36
Tabla 7 Características del Sensor LM35	37
Tabla 8 Características Motorreductor con caja reductora	38
Tabla 9 Características de Motor pasó a paso.....	39
Tabla 10 Diferencias entre servidor Virtual Privado y Servidor Dedicado	56
Tabla 11 Costo de los elementos tecnológicos	63
Tabla 12 Costo de los Recursos de Hardware	64
Tabla 13 Costo total de los Recursos del Software.....	64
Tabla 14 Costo total de los recursos humanos.....	64
Tabla 15 Costo total del prototipo	65
Tabla 16 Vulnerabilidades, Amenazas, riegos y categorías.	66
Tabla 17 Diccionario de datos – Base de datos control_secadero - tabla sensor_hyt.....	72
Tabla 18 Diccionario de datos – Base de datos control_secadero - tabla usuarios.....	72
Tabla 19 Diccionario de datos – Base de datos control_secadero - tabla clientes.....	73

Figuras

Figura 1 Placa de desarrollo Arduino Uno R3.....	30
Figura 2 Placa de desarrollo Raspberry Pi 3.....	31
Figura 3 Módulo WiFi ESP8266	33
Figura 4 Ethernet Shield	34
Figura 5 Sensor DHT11	35
Figura 6 Sensor DHT22.....	36
Figura 7 Sensor LM35	36
Figura 8 Motorreductor con Caja Reductora	37
Figura 9 Motor pasó a paso.....	38
Figura 10 Tornillo sin fin.....	40
Figura 11 Modelo Entidad Relación	81
Figura 12 Bosquejo prototipo control temperatura.....	86
Figura 13 Prototipo - Conexión Arduino UNO – Motor, Ventilador y Buzzer.....	87
Figura 14 Prototipo - Conexión Arduino UNO - Motor reductor.....	88
Figura 15 Dirección MAC e IP del servidor	89
Figura 16 Asignación de variables para conexión al servidor y envío de datos	90
Figura 17 Condición para la conexión hacia el servidor y envío de datos	90
Figura 18 Conexión Servidor.....	91
Figura 19 Conexión PDO a base de datos servidor	92
Figura 20 Inserción de datos a la base de datos del servidor	92
Figura 21 Conexión PHP con Arduino por el método GET	93
Figura 22 Verificación de datos en la base de datos del servidor	93
Figura 23 Modelo Vista Controlador.....	94
Figura 24 Archivos requeridos en el index.php	95
Figura 25 Archivo plantilla.controlado.php.....	95
Figura 26 Variable de SESSION	96
Figura 27 Control de ruta sistema.....	97
Figura 28 Formulario Ingreso al sistema	98
Figura 29 Controlador Ingreso Usuario	99
Figura 30 Comparación de la variable \$respuesta	99
Figura 31 Valores que están registrados en la tabla usuario.....	100
Figura 32 Alerta Bootstrap.....	100
Figura 33 Base de datos – Tabla Usuarios.....	101
Figura 34 Gestor de Usuarios	102
Figura 35 Usuarios - Maquetación.....	103
Figura 36 Usuarios - Controlador Mostrar Usuarios	103
Figura 37 Usuarios - Modelo Mostrar Usuarios	104
Figura 38 Usuarios - concatenar valores.....	105
Figura 39 Usuarios - Botón agregar usuario	105
Figura 40 Usuarios - Controlador Crear Usuario.....	106

Figura 41 Usuarios - Validación de la información en el controlador.....	107
Figura 42 Usuarios - Encriptación y creación del Modelo ingresar usuario.....	107
Figura 43 Usuarios - Modelo Mostrar Usuarios	108
Figura 44 Usuarios - Footer del modal agregar usuario	109
Figura 45 Usuarios - Validación de información Controlador Editar Usuario	110
Figura 46 Usuarios - Alerta Suave.....	111
Figura 47 Usuarios - Modelo Editar Usuario.....	112
Figura 48 Usuarios - Controlador Eliminar Usuario.....	112
Figura 49 Usuarios - Botón Eliminar usuario	113
Figura 50 Usuarios - Condición para borrar un usuario	113
Figura 51 Usuarios - Creación del Modelo Borrar Usuarios	113
Figura 52 Usuarios - Modelo Borrar Usuario	114
Figura 53 Clientes - Base de datos Tabla clientes	114
Figura 54 Clientes - Gestor de Clientes	115
Figura 55 Monitoreo tiempo real – Grafica temperatura	117
Figura 56 Monitoreo tiempo real – Grafica humedad	117
Figura 57 Monitoreo tiempo real – Grafica temperatura y humedad	118
Figura 58 Monitoreo Tiempo real - Conexión base de datos.....	118
Figura 59 Monitoreo Tiempo real - Requiriendo archivo de conexión	119
Figura 60 Monitoreo Tiempo real - Recargo de la página.....	120
Figura 61 Monitoreo Tiempo real - Consulta PHP para la gráfica temperatura	120
Figura 62 Monitoreo Tiempo real - Configuración yAxis para la gráfica de temperatura	121
Figura 63 Monitoreo Tiempo real - Series de la gráfica en paralelo temperatura y humedad ...	122
Figura 64 Monitoreo Tiempo real - Incluir archivos en Iframe para su visualización	122
Figura 65 Módulo de Estadísticas	123
Figura 66 Estadísticas - Botón daterange.....	124
Figura 67 Estadísticas - Botón Daterange Picker	124
Figura 68 Estadísticas - Configuración botón daterange	125
Figura 69 Estadísticas – LocalStorage	126
Figura 70 Estadísticas - LocalStorage Opción cancelar	126
Figura 71 Estadísticas - Capturando fecha "Hoy"	127
Figura 72 Estadísticas - Condición para fechaIncial y fechaFinal.....	128
Figura 73 Estadísticas - Controlador Rango Fechas	129
Figura 74 Estadísticas - Modelo Rango Fechas	130
Figura 75 Estadísticas - Array para agrupar datos	131
Figura 76 Estadísticas – Element.....	132
Figura 77 Estadísticas – Data.....	132
Figura 78 Estadísticas - Etiquetas para la gráfica Morris js.....	133
Figura 79 Informes – Reportes	134
Figura 80 Reportes – Conexión PDO	135
Figura 81 Informes - Reportes - Consulta para mostrar registros.....	135
Figura 82 Informes - Reportes - maquetación tabla de reportes	136
Figura 83 Informes - Reportes - Método GET fecha.....	137

Figura 84 Informes - Reportes - Consulta SQL para ingresar los registros	137
Figura 85 Informes - Reportes - Consulta SQL para el promedio	138
Figura 86 Informes - Reportes - Consulta SQL para los datos máximos y mínimos	138
Figura 87 Informes - Reportes – Registros	139
Figura 88 Informes - Reportes - Promedio temperatura y humedad	139
Figura 89 Informes - Reportes - Datos Máximos y Mínimos	140
Figura 90 Módulo de Inicio	141
Figura 91 Inicio - Vista Rol de Administrador	142
Figura 92 Inicio - Vista Rol de Especial	143
Figura 93 Inicio - Inicio - Vista Rol de Especial - Administrador de usuarios	143
Figura 94 Inicio - Vista Rol de Especial - Administrador de Clientes	144
Figura 95 Inicio - Vista Rol de Cliente	144
Figura 96 Alerta Generada por el sistema.....	145
Figura 97 Resultado Encuesta - Pregunta 1	159
Figura 98 Resultado Encuesta - Pregunta 2	160
Figura 99 Resultado Encuesta - Pregunta 3	161
Figura 100 Resultado Encuesta - Pregunta 4	162
Figura 101 Resultado Encuesta - Pregunta 5	163
Figura 102 Resultado Encuesta - Pregunta 6	164
Figura 103 Resultado Encuesta - Pregunta 7	164
Figura 104 Resultado Encuesta - Pregunta 8	165

Introducción

Actualmente el auge de la tecnología se ha visto muy notoria entre la población, debido al avance a gran escala en la última década. En la actualidad se está implementando el internet de las cosas IOT (Internet Of Things), concepto que se refiere a la conexión a la red de todo tipo de objetos (televisores, coches, libros, relojes, vestidos, drones, etc.), lo que multiplicará sus aplicaciones y transformará tanto el uso que se da a los mismos como de muchos aspectos de la vida humana en general (Robotsia, 2015).

Este proyecto se fundamenta en implementar un sistema de control y monitoreo de temperatura, para lo cual se diseñó un prototipo funcional con una excelente calidad y un presupuesto relativamente bajo; desarrollado mediante las tarjetas de programación Arduino, el IDE de Arduino y algunos lenguajes de programación como: PHP, MySQL, Bootstrap 4, CSS3 y JavaScript. Este método tiene como objetivo principal mantener la temperatura ideal y no exceder los 50°C en el sistema de secado al sol, para ello se complementa con la implementación de Motores Reductores y un Ventilador que tiene como tarea reducir los altos niveles de temperatura en el sistema.

Toda la información recolectada por los sensores de temperatura y humedad se sube a una Base de datos y luego esta información es visualizada en una página web; donde el usuario del proyecto puede visualizar los datos en tiempo real, generar consultas y reportes. A demás, el acceso al sistema de control y monitoreo podrá realizarse desde cualquier otro dispositivo diferente al que se encuentra implementado el proyecto.

Capítulos

Capítulo 1. Aspectos generales

1.1 Selección y definición del tema de investigación

Enfocándonos principalmente en la IOT (Internet Of Things) es un tema de gran interés ya que este tipo de tecnología se está implementado a nivel mundial, donde se menciona:

Tan solo en 2015 alrededor de 10 mil millones de dispositivos se conectaron a Internet, mientras que para el 2020 se espera que habrá 34 mil millones de dispositivos, de los cuales 24 mil millones representarán al IoT y solo 10 mil millones serán dispositivos tradicionales de computación (por ejemplo, teléfonos inteligentes, tabletas, smartwatches, etc.) (ACIS, 2016).

La información que se puede encontrar actualmente sobre esta tecnología es muy grande y avanza con gran rapidez, debido a que la mayoría de sus proyectos son de software libre o código abierto (Open Source). En términos generales se puede consultar información en cualquier motor de búsqueda de interés como: Google, Bing, Yahoo!, YouTube, etc.

El desarrollo de esta investigación está ligada con experiencias vividas de las altas temperaturas en los sistemas de secado al sol (secaderos hechos por el campesino), y de la misma manera se ha tomado este tema en particular con el objetivo de obtener resultados personales en cuanto al procedimiento que se requiere para la implementación en un secadero de mayor tamaño y gran cantidad de volumen de café. De igual forma en los resultados generales se resalta la importancia de la aplicación de las nuevas tecnologías con el objetivo de hacerle a los caficultores una vida más práctica y más cómoda, gracias a la interconexión de la tecnología para el mejoramiento de la calidad de vida.

1.2 El problema de investigación

Durante muchos años la forma de realizar el secado del Café, se ha estandarizado por ser a través de secaderos solares hechos manualmente por el hombre el cual están contruidos por: madera, plástico calibre 6, poli-sombra y cisco. También emplean Silos (secado mecánico) el cual se recomienda para las fincas con producciones de café mayores a 500 arrobas. Para este tipo de secado mecánico se trabaja a carbón, diésel, gas natural o propano y cisco. La temperatura máxima a la que pueden someterse los granos de café, sin experimentar daños irreversibles en la estructura interna, es de 50°C, en las condiciones atmosféricas de la zona cafetera colombiana, ese calentamiento corresponde a una humedad relativa por debajo del 20% (GONZALEZ, C., SANZ, J., & Oliveros, C. E. , 2014).

De acuerdo a estos criterios, no importa cual método se maneje para el secado del café siempre se va a tener que utilizar un operario, quien estará a cargo de la temperatura y el volteado del café haciendo irregular la calidad del producto ya que no se tiene una cifra específica de acaloramiento, ni tampoco un tiempo predeterminado para obtener el producto final del grano de café y que esté listo para su comercialización.

Desde otro punto de vista se comprende que el campesino al estar expuesto a altas temperaturas puede sufrir problemas de salud. “Cuando la temperatura corporal excede lo óptimo para su funcionamiento (36,6° a 37°C), el cuerpo reacciona para eliminar el exceso de calor principalmente de dos maneras: Aumento de la circulación de la sangre y más sudor” (Ramos Fuentes, 2017). Cuando los mecanismos fisiológicos de eliminación de calor son insuficientes y persiste la agresión térmica, la temperatura interna del organismo aumenta hasta que se produce

el golpe de calor, con pérdida de consciencia, estado de coma y en algunos casos la muerte

(Remón, 2011).

En este propósito, los efectos de salud pueden clasificarse en:

- A. Alteraciones Cutáneas: Se dividen en dos grupos Erupción y quemaduras.
- B. Alteraciones Sistémicas: estas afectan al sistema, encontramos el golpe de calor, agotamiento por calor, deshidratación y déficit salino.

1.2.1 Planteamiento del problema (Síntomas, causas, pronóstico y control pronóstico)

Se realiza una “descripción de la “situación actual” que caracteriza al “objeto de conocimiento” (síntomas y causas) identificación las situaciones futuras al sostenerse la situación actual (pronóstico). Presentación de alternativas para superar la situación actual (control al pronóstico)” (Bermúdez & Rodríguez, 2013).

Tabla 1 Síntomas, Causas, Pronóstico y Control pronóstico

Síntomas	Causas	Pronóstico	Control al Pronóstico
Se presentan una leve contaminación Ambiental y Auditiva	Producida por los Silos (secado mecánico) para el secado del café.	Podría seguir aumentando la contaminación atmosférica y auditiva. A demás de tener una baja calidad de café.	Aplicar el un sistema para el secado amigable al medio ambiente y con una rapidez casi igual a la de los Silos.
Una tasa muy Baja calidad del Café	Mal manejo sobre el tratamiento de enjuague y secado del café.	Seguirían los caficultores más pequeños con una desproporción en su producto y no se obtienen los resultados esperados.	Se podría tener unas medidas específicas para lograr controlar la temperatura del café.
Una situación económica difícil, debido al precio del producto a nivel nacional	Debido a que en la mayoría de los caficultores no emplean técnicas para mejorar el producto final del café.	Se mantendría una monotonía muy baja sobre el precio del mismo y esto seguiría a raíz que no se logre un esfuerzo para mejorar su producto.	Aplicar técnicas o instrumentos especializados para lograr obtener un excelente nivel de productividad.
Poco conocimiento en	Es producido por falta de información, ya	A futuro se pensaría que el avance tecnológico sería	Brindar asesorías a las personas que menos

la zona rural sobre la tecnología	que es una nueva tecnología que se está implementando a nivel nacional.	un poco más lento y a raíz del mismo la disminución en proyectos a implementar.	obtengan información de la misma y así ir contextualizando a la sociedad sobre estos nuevos conocimientos.
Problemas en el manejo del Sistema a desarrollar	Porque los caficultores o campesinos no poseen los conocimientos necesarios para el manejo del sistema.	Se podría presentar fallas en el sistema además de obtener datos erróneos sobre el objetivo del proyecto.	Proporcionar una capacitación a los caficultores sobre el manejo del sistema y todas sus características.

Autor: Hector Fabio Duran Montes

1.2.2 Formulación del Problema

¿Por medio del prototipo funcional se puede controlar y monitorear en tiempo real la temperatura y humedad de una forma precisa y exacta en el sistema de secado al sol para el café?

1.2.3 Sistematización del problema.

- ¿Los caficultores tienen conocimientos sobre el manejo de dispositivos como: Computador, Tablet y Celular Android?
- ¿En todas las fincas tendrán conexión a internet?
- ¿Cuál será la mejor forma para el caficultor comercializar el café? (Mojado o Seco)
- ¿Se puede determinar la humedad y temperatura del café de manera exacta?
- ¿De qué manera se puede controlar la temperatura en el prototipo funcional?
- ¿Qué alternativas se tendrían que implementar para los caficultores que no poseen conexión a internet?
- ¿El aplicativo se podrá acceder desde cualquier otro dispositivo en el que se encuentra implementado?
- Dado el caso en que falle la conexión a internet ¿Qué alternativas de contingencia se tienen para el proceso de secado del café?

1.3 Línea de investigación

Se aplicara la línea de *Gestión de Sistemas* está orientada a integrar, planificar y controlar los aspectos técnicos, humanos, organizativos, comerciales y sociales del proceso completo, empezando con el análisis del dominio del problema, continuando con el diseño de alternativas de solución y finalizando con la operatividad de un sistema. La gestión de sistemas incluye también procesos que abarcan la planificación de actividades, metas, responsables, indicadores de eficiencia, eficacia y efectividad.

El objetivo de esta línea de investigación es Apoyar el desarrollo productivo, tecnológico y social empresarial a través del análisis, diseño, implementación o administración de sistemas de información y las TIC que estén basados en la planificación, dirección, control, evaluación y realimentación de actividades procedimentales (UNAD, 2011).

1.4. Objetivos de la investigación

En base a lo expuesto anteriormente, esta investigación considera la importancia de llevar a cabo una búsqueda exhaustiva sobre las diferentes metodologías que se complementan en el momento de aplicar el diseño e implementación del aplicativo. Para ello El tipo de software definido para este proyecto es aquel que permite realizar aplicaciones que sean independientes de las acciones tomadas por el usuario, es decir un tipo de software que en algunas oportunidades pueda de manera propia tomar ciertas decisiones en el transcurso de la ejecución del aplicativo, además que permita interactuar con el usuario, analizar, y generar consultas sobre la información recolectada. En base a estos requerimientos se define que el tipo de software que mejor se ajusta a esta definición, es el de inteligencia artificial, para lo cual permite a través de procesos lógicos encontrar una solución heurística a ciertos problemas, e integrar en el software sistemas que

piensen como humanos de manera artificial, y que actúen racionalmente en ciertos puntos sobre el procedimiento que está llevando a cabo el sistema.

1.4.1 Objetivo General

Diseñar un prototipo funcional para el control y monitoreo de la temperatura y humedad del ambiente, registrando en tiempo real los datos obtenidos en el sistema de secado al sol para el café.

1.4.2 Objetivos Específicos

- Recolectar y analizar la información suficiente que permita la identificación del problema a resolver
- Analizar cada una de las metodologías y normas que rigen la elaboración de un software.
- Realizar la planificación del proyecto que permita establecer parámetros en las 4 gestiones que por estándar exige la creación del sistema.
- Aplicar las mejoras que sean necesarias al momento de la planificación del diseño del Software y el prototipo.
- Implementar nuevas tecnologías que permitan el correcto aprovechamiento del software a diseñar.

1.5 Justificación de la investigación

El manejo para el secado del café manualmente trae consigo mismo irregularidades que de una u otra manera influye directamente en la calidad del producto y el caficultor se verá afectado cuando vaya a comercializarlo, en conclusión no obtendrá el resultado esperado en comparación con un producto de alta calidad.

Cuando se utiliza un operario para el proceso de secado tiene como objetivo principal revolver el café cada determinado tiempo y realizar varias pruebas sobre la calidad del producto donde determina si está listo para su comercialización. Como consecuencia estará expuesto a un calor excesivo en el secadero solar, según (Dir.ca.gov, 2012) “las enfermedades relacionadas con el calor son un problema de salud grave. Si no se toman precauciones a tiempo, las enfermedades causadas por calor pueden afectar la salud de los trabajadores seriamente e incluso causar la muerte.” Por eso es importante que los caficultores y empleadores deban conocer los síntomas de las enfermedades causadas por las altas temperaturas y así ellos poder realizar medidas de precaución para evitar muertes o daños en la salud.

De esta manera para prevenir enfermedades causadas por las altas temperaturas es necesario saber cómo el cuerpo humano se enfría cuando se ha sobrecalentado.

Según (Dir.ca.gov, 2012) el cuerpo humano necesita mantener una temperatura interna de unos 98.6 grados Fahrenheit (37 grados Celsius) para poder funcionar óptimamente. Cuando el cuerpo se calienta por encima de esta temperatura, reacciona para eliminar el exceso de calor. El cuerpo humano hace esto principalmente de dos maneras:

- ✓ Aumento de la circulación de la sangre: Cuando estamos expuestos a un calor extremo, nuestro corazón late más rápido y bombea más sangre. Hay un aumento en la velocidad y

la cantidad de sangre que circula cerca de la piel. A medida que la sangre se acerca a la superficie de nuestra piel, el calor se libera al medio ambiente. Por eso la gente de piel clara se ve enrojecida.

- ✓ Más sudor: Al mismo tiempo que aumenta la circulación de la sangre, nuestro cuerpo tiende a sudar más. Al evaporarse el sudor, se libera calor de nuestro cuerpo.

Teniendo en cuenta esta información, se puede concluir que el operario en el proceso de secado estará expuesto a temperaturas elevadas a los 37°C ya que en el secadero solar se pueden manejar temperaturas hasta sobrepasar los 50°C y esto podría ocasionar enfermedades de la salud, sino se toman las prevenciones adecuadas para el proceso que está llevando. Algunas de las enfermedades causadas por el excesivo calor son:

Tabla 2 Enfermedades causadas por las altas temperaturas

Enfermedad	Descripción	Tratamiento
Sarpullido	Áreas de la piel irritadas e inflamadas, que pueden estar enrojecidas, tener picazón, escamas, costras o ampollas. Esto sucede porque las glándulas del sudor se tapan debido al calor, la humedad y el sudor excesivo.	<ul style="list-style-type: none"> • Mantenga la piel limpia y seca. • Descanse en un lugar fresco. • Beba agua. • Cámbiese de ropa con frecuencia para mantenerse seco.
Calambre	Calambres musculares dolorosos, por lo general en las piernas o abdomen (barriga) causados por la pérdida excesiva de sal a través del sudor. Esto es una señal de que una enfermedad causada por el calor se está desarrollando.	<ul style="list-style-type: none"> • Tome descansos en un ambiente más fresco. • Beba agua • Quítese cualquier Equipo de Protección Personal, y remueva o afloje la ropa apretada. • Si es posible recuéstese en un lugar fresco.
Agotamiento	Cuando los líquidos no se reponen después de mucho trabajo, hay pérdida excesiva de agua y de sal por medio del sudor. La persona puede sentirse cansada, débil, mareada, y con la piel muy húmeda. Esta es una condición seria.	<ul style="list-style-type: none"> • Haga que la persona descanse en un lugar fresco y que beba agua (si la persona no tiene náuseas). • Si es posible, haga que la persona se acueste con las rodillas levantadas. • Afloje la ropa • Busque ayuda Medica

Insolación	<p>Esta es una condición potencialmente mortal en la cual la temperatura interna del cuerpo se eleva por encima de 105°F (41 °C) y en la que las funciones vitales se atrofian, incluyendo las funciones mentales. Sin atención médica inmediata la insolación puede resultar en daño cerebral permanente o muerte.</p>	<ul style="list-style-type: none">• Notifique a su supervisor• EMERGENCIA: BUSQUE AYUDA MEDICA INMEDIATAMENTE!!• Coloque la persona en un lugar fresco y afloje la ropa.• Ponga toallas mojadas y húmedas debajo de las axilas y la ingle, y use un ventilador para crear circulación del aire.• Evite el frío extremo, porque el cuerpo puede entrar en shock.• EVITE llevar a la persona al hospital en un carro que esté caliente.
------------	--	---

Autor: (Dir.ca.gov, 2012)

Analizando el proceso que lleva el secado del café con los silos, ha de notarse que son equipos que producen una contaminación ambiental y auditiva esto porque son de un gran volumen y su funcionamiento así lo requiere. Por este motivo es importante reducir los niveles de contaminación que se ejerce con los silos y empezar a emplear sistemas para el secado del café de bajo presupuesto, con facilidad de uso, elementos tecnológicos de calidad y que logre en realizar un trabajo autónomo y sistematizado al momento de llevar a cabo este proceso.

Por otra parte con la implementación de este sistema mejora significativamente el proceso de secado, ya que realiza un control para la temperatura y humedad relativa del ambiente que se obtiene en el proceso de secado para el café. Además de aportar nuevas tecnologías que tendrán como objetivo hacerle a los caficultores una vida más práctica y más cómoda, gracias a la interconexión con la tecnología y el mejoramiento de la calidad de vida.

1.5.1 Motivación teórica

Arduino es la programación de un microcontrolador. Esto era algo más de los ingenieros electrónicos, pero Arduino lo ha extendido a todo el público. Arduino ha socializado la tecnología. Programar Arduino consiste en traducir a líneas de código las tareas automatizadas que se quieren hacer leyendo de los sensores y en función de las condiciones del entorno programar la interacción con el mundo exterior mediante unos actuadores.

Arduino proporciona un entorno de programación sencillo y potente para programar, pero además incluye las herramientas necesarias para compilar el programa y “quemar” el programa ya compilado en la memoria flash del microcontrolador. Además el IDE nos ofrece un sistema de gestión de librerías y placas muy práctico. Como IDE es un software sencillo que carece de funciones avanzadas típicas de otros IDEs, pero suficiente para programar (Aprendiendo Arduino, 2013).

La parte fundamental de la estructura del Arduino es el Sketch, ya es el programa como tal y su extensión es .ino. Este documento se compone de dos partes fundamentales de programación estas son: setup () y loop (), el primero es donde se recopilan las diferentes configuraciones y el segundo es donde estará integrado el programa que se ejecutara. Estas dos funciones son obligatoriamente necesarias para que el programa funcione.

El lenguaje de programación de Arduino es C++. No es un C++ puro sino que es una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad para usar con GCC compilador de C y C++ (Aprendiendo Arduino, 2013). Además de ello Arduino proporciona librerías para toda clase de proyectos que estén en su imaginación y en el alcance que proporciona esta tecnología.

De esta manera al conocer el potencial de Arduino y saber que tiene una interconexión tecnológica con los diferentes lenguajes de programación, se diseñó una propuesta para el prototipo del control de temperatura en el proceso de secado para el café. Esta propuesta conlleva muchos interrogantes, diferentes enfoques teóricos, investigaciones, alternativas de solución para el prototipo, entre otras. Siempre teniendo en cuenta el nuevo conocimiento adquirido durante el proceso de desarrollo del proyecto, el cual complementan el conocimiento inicial y de la misma manera encontrando nuevas explicaciones para avanzar significativamente en la solución óptima del prototipo.

El proyecto se divide en 3 enfoques principales de investigación, el primero consta del diseño del prototipo físico el cual es elaborado con Arduino y componentes de electrónica. El segundo es la conexión que se establece entre la placa de Arduino con los lenguajes de programación y gestores de bases de datos. El tercero es el diseño y elaboración del sitio web donde se obtendrán los resultados enviados por el prototipo, con el objetivo de analizar la información y presentarla al usuario de una forma eficaz y eficiente.

1.5.2 Motivación metodológica

En el modelo de esta investigación como primera instancia se realizaron una recolección de información para los requerimientos técnicos que contempla el proyecto y de igual forma realizar una planificación acertada sobre las necesidades que el caficultor requiere. Para ello se aplicaron las técnicas de recolección de datos como: Observación directa, videos, entrevistas o encuestas. Estas técnicas se aplicaron a una comunidad específica, donde se realizó una delimitación geográfica sobre la información que se presentaría para el desarrollo del proyecto.

Esta técnica de recolección de datos podrá visualizarse en el **Anexo 1**

1.5.3 Motivación práctica

Inicialmente el diseño de esta propuesta de proyecto aplicado se realiza con el objetivo de incrementar los conocimientos adquiridos durante los 5 años de estudios universitarios y de la misma manera se toma como un reto académico en donde se darán a conocer las capacidades que tiene el estudiante para emplear diferentes tecnologías y proyectos de grandes magnitudes, además de contribuir para obtener el título de ingeniero de sistemas.

Por otra parte este proyecto tiene una razón social la cual consiste en contribuir a diseñar un prototipo funcional que será implementado de forma efectiva para el proceso de secado del café y en el futuro quizás se pueda emplear directamente en los secaderos de las fincas cafeteras.

1.6 Aspectos Metodológicos

De acuerdo con los razonamientos que se han venido realizando, en primera instancia se realizó la recolección de la información mediante la técnica de encuesta y la observación directa. Seguidamente los resultados obtenidos son analizados y representados mediante gráficas de barras y circular, con el objetivo de identificar la problemática que con lleva el proceso de secado manualmente y los diferentes recursos que posee el caficultor en cuanto a la implementación de un nuevo sistema para este proceso. De esta manera se originó el desarrollo del prototipo funcional para el control y monitoreo de la temperatura sobre el proceso de secado del café.

1.6.1 Tipo de Estudio

En este proyecto se utiliza el estudio exploratorio y Descriptivo, debido a que este es empleado cuando el fenómeno ha sido poco estudiado o es completamente novedoso y acuden a técnicas específicas en la recolección de información, como la observación, las entrevistas y los cuestionarios.

Los estudios exploratorios permiten realizar una aproximación a fenómenos, por tanto, desconocidos, con el fin de aumentar el grado de familiaridad frente a éstos. Las exploraciones se centran en descubrir, es decir se busca contribuir con ideas respecto a la forma correcta de abordar una investigación en particular. Los estudios descriptivos tiene el propósito de describir sistemáticamente los hechos y características de una población dada o de un área de interés, su finalidad es identificar, reconocer, describir y mostrar, para ello pueden utilizar tratamientos estadísticos o categorizar la información en matrices de análisis, buscando siempre presentar en forma detallada las características de su objeto de estudio (Sosa, 2016).

1.6.2 Método de investigación

Se realizó la recolección de información a través de la técnica encuesta y observación directa en la población de la vereda San Marcos (Acevedo – Huila) para la cual se realizaron encuestas a los caficultores que poseen fincas pequeñas (estas son las que tienen plantaciones de café con menos de 5 hectáreas) además de observar directamente como es el proceso manual que realiza el campesino para lograr obtener un resultado final con respecto a su producto.

Según (cenicafe) el valor a pagar al caficultor por su café se calcula mediante el factor de rendimiento en trilla, que es la cantidad de kilogramos de café pergamino seco necesarios para

obtener un saco de 70 kilogramos de café almendra. El precio obtenido en la comercialización depende de la calidad de café que se lleve al punto de compra.

El factor de rendimiento en trilla óptimo es de 92,8 kilogramos de café pergamino. A menor factor de rendimiento mayor precio, y a mayor factor de rendimiento menor precio.

El Café pergamino para la venta debe tener las siguientes características:

- ✓ Humedad entre el 10 y el 12%.
- ✓ Olor fresco característico. No se acepta café con olor a reposado, moho, tierra, vinagre, petróleo, etc.
- ✓ Color uniforme.
- ✓ La almendra no puede tener granos flojos, cardenillos y vinagres.
- ✓ El café debe estar libre de todo insecto vivo o muerto.

Una vez se ha identificado el problema e investigado los requerimientos establecidos por la FNC para lograr obtener unos posibles estándares de calidad del café se diseña el circuito del prototipo para el sistema de secado del café según lo expuesto en los apartados anteriormente mencionados. Por último se realizan pruebas de control y monitoreo de temperatura y humedad en el software que se ha implementado para el proyecto, además de las pruebas de funcionalidad, usabilidad, eficacia y seguridad que se le debe realizar al software por medio del usuario.

1.6.3 Fuentes y técnicas para recolección de información

Para esta investigación se aplica las **fuentes primarias**, donde la técnica de recolección de datos más importante y relevante fue la encuesta realizada a las personas que viven en la comunidad que se ha delimitado para el prototipo funcional. A demás se realizaron entrevistas y

técnicas de observación con referente al proceso manual que realiza el caficultor para llevar a cabo el secado del café.

1.6.4 Tratamiento de la Información

Toda la información recolectada a través de las encuestas es presentada en forma escrita, tabulada y mediante gráficos para una mayor pre-visualización de la información como lo son:

Gráficos de Barras, Polígonos, Tortas, etc. (esta es una técnica de recolección de datos primaria)

Estos resultados se podrán visualizar en el **Anexo 2**

Capítulo 2. Marco de referencia de la investigación

2.1 Marco teórico.

Dado que este trabajo está enfocado en diseñar e implementar un prototipo funcional de control y monitoreo para el sistema de secado del café, resulta fundamental tener una definición más amplia sobre los parámetros exigidos para el café de Colombia.

Según (ColCoffeeInsights, 2012) el 96% del total de los caficultores del país tienen plantaciones de café de menos de 5 hectáreas y el tamaño promedio de los cultivos de café en el país apenas supera 1.6 hectáreas. Desde un punto de vista técnico, las prácticas inadecuadas durante el secado del café pueden causar 10 de los 14 defectos establecidos por los estándares de calidad aplicados al Café de Colombia. Estos parámetros exigen un contenido de humedad entre 10 – 12% en los granos y una temperatura que no exceda los 50 °C, con el fin de asegurar un control y un monitoreo preestablecido por el software que se desarrolló.

El secado al sol es un método económico y ambientalmente amigable para cumplir o superar los estándares mínimos de calidad del café de Colombia. Es el más usado por los pequeños productores en Colombia, quienes producen cerca del 70% de la cosecha nacional (ColCoffeeInsights, 2012).

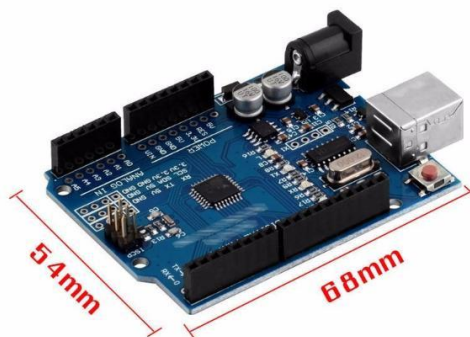
Siguiendo estos lineamientos anteriormente plasmados se analiza que el valor de la humedad está en un rango entre el 10% - 12% y la temperatura por debajo de los 50 °C, resulta este ser el parámetro adecuado para lograr obtener un control y monitoreo lo suficientemente favorable para el desarrollo del software.

Placas de Desarrollo

Cuando nos enfrentamos a un proyecto con estas características, es fundamental contar con un elemento indispensable en el desarrollo del software como lo es la placa de desarrollo. En el mercado se pueden encontrar varias placas que a raíz de sus características son muy opcionadas para la implementación y solución de la propuesta de estudio. Posteriormente se hace un análisis de las placas que tengan como característica principal conexión a internet, popularidad, fácil acceso de información, librerías y recursos de adaptabilidad.

Arduino: “Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos” (Arduino, s.f.). El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Además cabe resaltar que el software se puede descargar gratuitamente.

Figura 1 Placa de desarrollo Arduino Uno R3



Fuente: Mactronica [fotografía]. Electrónica: Arduino Uno R3 ATMEGA328P Montaje Superficial. Disponible en internet: <https://www.mactronica.com.co/arduino-uno-r3-atmega328p-montaje-superficial-96165481xJM>

¿Porque elegir Arduino? Simplemente porque Arduino simplifica el proceso de trabajo con microcontroladores y se obtienen grandes ventajas como: Su costo económico, Multiplataforma, Entorno de programación simple y clara, código abierto y software extensible y por ultimo código abierto y hardware extensible. (Arduino, s.f.)

A continuación se mencionan las placas de desarrollo de Arduino más conocidas.

1. Arduino Uno R3
2. Arduino Mega
3. Arduino Ethernet
4. Arduino Leonardo
5. Arduino Micro
6. Arduino Nano

Raspberry PI: es una placa computadora (SBC) es de mayor costo que un Arduino uno, se podría decir que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012 (Tecnologías, 2013).

Figura 2 Placa de desarrollo Raspberry Pi 3



Fuente: Vistronica [fotografía]. Electrónica: Raspberry Pi 3 Modelo B+. Disponible en internet: <https://www.vistronica.com/board-de-desarrollo/raspberry-pi/raspberry-pi-3-modelo-b-detail.html>

Actualmente existen varios modelos en el mercado de la Raspberry pi y cada una incorpora características lo suficientemente precisas para que se establezcan como únicas en sus versiones más actuales, entre estas se encuentran el Modelo A, 2 Modelo B y 3 Modelos B.

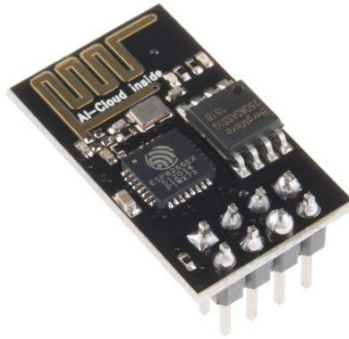
Tabla 3 Características específicas de las diferentes placas Raspberry pi

	Modelo A	2 Modelo B	3 Modelo B
Soc	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837
CPU	ARM1176JZF-S 700 MHz	ARM Cortex-A7 900 MHz Quad-core	QUAD ARM Cortex -A53 1.2 GHz
GPU	VideoCore IV	VideoCore IV	VideoCore IV
RAM	256 Mb	512 Mb	1 Gb
USB	1	2	4
Video	RCA, HDMI	RCA, HDMI	Jack, HDMI
Audio	Jack ,HDMI	Jack ,HDMI	Jack ,HDMI
Boot	SD	SD	MicroSD
Red	-	Ethernet 10/100	Ethernet 10/100, Wi_, BT
Consumo	300mA/1.5w/5V	800mA/4w/5V	2.5A/12.5w/5V
Precio	25 \$	35 \$	35 \$

Autor: (Woratek, 2012)

Módulo Wifi ESP8266: es un chip WiFi de bajo coste con protocolos TCP/IP completa y capacidad de MCU (Micro Controller Unit) producida por el fabricante chino Espressif Systems, con sede en Shanghai. Este pequeño módulo permite a los microcontroladores conectarse a una red WiFi y realizar conexiones TCP/IP sencillas utilizando comandos de tipo Hayes (Aprendiendo Arduino, 2013). El precio es muy bajo, ya que en el mercado colombiano se puede encontrar en las tiendas online por valor de \$ 11.000 Pesos Colombianos.

Figura 3 Módulo WiFi ESP8266



Fuente: Mactronica [fotografía]. Electrónica: Modulo WIFI ESP8266. Disponible en internet: <https://www.mactronica.com.co/modulo-wifi-esp8266-43672064xJM>

Algunas características de este pequeño modulo se pueden encontrar en el datasheet, el cual se expresan en el siguiente cuadro.

Tabla 4 Características Técnicas del módulo ESP8266

Parámetro	Condiciones Técnicas
Voltaje de operación	3.3 V - 3.6 V
Corriente de operación	80 mA
Protocolos WiFi	802.11 b/g/n
Rango de frecuencia	2.4 GHz - 2.5 GHz
Consumo Deep Sleep	< 10 μA

Autor: (Laborda, 2016)

Ethernet Shield: Da la capacidad de conectar un Arduino a una red ethernet. Es la parte física que implementa la pila de protocolos TCP/IP. Está basada en el chip ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de soportar TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto Ethernet. Me permitirá escribir sketches que se conecten a internet usando la Shield (Aprendiendo Arduino, 2016).

Figura 4 Ethernet Shield



Fuente: Mactronica [fotografía]. Electrónica: Ethernet Shield W5100. Disponible en internet:
<https://www.mactronica.com.co/ethernet-shield-w5100-43990642xJM>

Características

- Posee un conector RJ45
- Usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100
- El botón de reset en la Shield resetea ambos, el W5100 y la placa Arduino.
- Posee LEDs para información como: ON, LINK, 100M, RX, TX y COLL

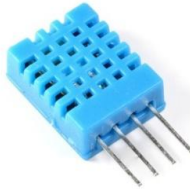
Sensores de humedad y temperatura

Por otra parte el sensor de temperatura y humedad juega un papel fundamental en la búsqueda para mejorar el prototipo de control y monitoreo del secado al sol del café. Gracias a este se obtendrá una información detallada del proceso que se está llevando a cabo, el cual servirá de gran ayuda para que el caficultor logre reducir los defectos causados por un mal procedimiento en el momento de realizar el secado del café.

Para ello se ha tenido en cuenta algunas posibilidades de sensores que de una u otra manera cuentan con las especificaciones necesarias para ser implementado en el desarrollo del prototipo funcional. Entre estos se encuentran el DHT11, DHT22 y el LM35.

Sensor DHT11: lo primero que se puede apreciar de este sensor es que es digital y a demás incorpora un sensor de humedad. Internamente realiza la conversión a grados centígrados y su mayor fuerte quizás sea su calidad pero por el contrario, solo se puede leer la temperatura cada dos segundos, tiene un rango de muestras por segundo muy bajo. Arduino incorpora una librería, dht.h, que permite leer toda la gama de sensores DHT. Esta librería facilita mucho la lectura de estos tipos de componentes (Del valle Hernandez, s.f.).

Figura 5 Sensor DHT11



Fuente: Mactronica [fotografía]. Electrónica: Sensor De Humedad Y Temperatura DHT11. Disponible en <https://www.mactronica.com.co/sensor-de-humedad-temperatura-dht11-107474339xJM>

El único problema que se puede encontrar es el rango de temperaturas ya que está comprendida entre 0° C a 50° C. A continuación se exponen las características de este sensor.

Tabla 5 Características del Sensor DHT11

Sensor DHT11	
Voltaje de operación	3 V a 5,5 V
Rango de temperaturas	0° C a 50° C
Precisión	±2° C
Resolución	8 bit
Muestras/segundo	0,5
Costo	\$6.000 Pesos COP

Autor: (Omniblug, 2013)

Sensor DHT22: este sensor es el hermano mayor del sensor DHT11, este sensor mide la humedad y temperatura en donde se resalta su alta fiabilidad y su estabilidad en las diferentes mediciones digitales que realiza. El rango abarca temperaturas bajo cero, tiene una precisión

mayor y la resolución es de 16 bit. A continuación se exponen las características que lo conforman.

Figura 6 Sensor DHT22



Fuente: Mactronica [fotografía]. Electrónica: Sensor De Humedad Y Temperatura DHT22. Disponible en <https://www.mactronica.com.co/sensor-de-humedad-temperatura-dht22-50562202xJM>

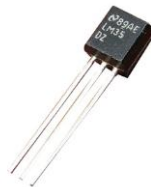
Tabla 6 Características del Sensor DHT22

Sensor DHT22	
Voltaje de operación	3,3 V a 6 V
Rango de temperaturas	-40° C a 80° C
Precisión	±0,5° C
Resolución	16 bit
Muestras/segundo	0,5
Costo	\$16.000 Pesos COP

Autor: (Omniblug, 2013)

Sensor LM35: este sensor está calibrado directamente en grados Celsius (centígrados) y no se debe hacer ningún tipo de conversión ni de calibración externa. La salida es analógica y se medirá con una entrada analógica de Arduino. El gran problema de este sensor es que realmente solo se puede medir temperaturas entre 2° C y 150° C a no ser que se utilicen voltajes negativos

Figura 7 Sensor LM35



Fuente: Vistronica [fotografía]. Electrónica: LM35 Sensor de Temperatura. Disponible en <https://www.vistronica.com/sensores/temperatura/lm35-sensor-de-temperatura-detail.html>

Esto es un problema cuando trabajamos con Arduino ya que la placa no suministra voltajes negativos y se debe utilizar algún circuito externo para conseguir este rango de tensiones. Se pueden encontrar otro tipo de soluciones en Internet pero lo único que se va a conseguir es que se pierda la linealidad en la temperatura y por lo tanto se obtendrá valores erróneos. La solución es alimentar con un voltaje negativo o utilizar otro tipo de sensor. A continuación se exponen las características principales de este sensor (Del valle Hernandez, s.f.).

Tabla 7 Características del Sensor LM35

Sensor LM35	
Voltaje de operación	4 V a 30 V
Rango de temperaturas	-55° C a 150° C
Precisión	±0,5° C
Conversión	10 mV / °C
Tiempo de respuesta (100%)	4 Minutos
Offset	0 V
Costo	\$6.000 Pesos COP

Autor: (Hetpro, 2017)

Motores

Motorreductor: es un reductor de velocidad con un motor directamente conectado. Un motorreductor no utiliza un adaptador campana y cople para montar el motor, el motor es de hecho parte del reductor de velocidad. Una primera ventaja del motorreductor comparado con otra combinación motor y reductor, es la longitud. Moto reductores son compactos en tamaño (Energía controlada, s.f.).

Figura 8 Motorreductor con Caja Reductora



Fuente: Mactronica [fotografía]. Electrónica: Motorreductor Plástico 6V 0.5KG/cm 220RPM 1:48. Disponible en <https://www.mactronica.com.co/motorreductor-plastico-6v-05kgcm-220rpm-148-76064727xJM>

Tabla 8 Características Motorreductor con caja reductora

Modelo	TGP01D-A130
Motor DC	-Imán permanente -Escobillas de carbón -Eje biaxial
Voltaje de operación	3V~6V
Corriente de operación	100mA (sin carga) 200mA (con carga)
Caja reductora (piñonería plástica)	1:48
Velocidad	9800rpm (sin carga) 220rpm (con carga)
Torque	0.5KgrCm
Tamaño	6.92cm x 2.25cm x 1.88cm
Peso	35 gramos
Precio	\$ 7.000 COP

Autor: (Vistronica, s.f.)

Motor Paso a Paso: (también llamado stepper) es un dispositivo electromagnético que convierte impulsos eléctricos en movimientos mecánicos de rotación. La principal característica de estos motores es que se mueven un paso por cada impulso que reciben. Normalmente los pasos pueden ser de 1,8° a 90° por paso, dependiendo del motor. Son motores con mucha precisión, que permiten quedar fijos en una posición (como un servomotor) y también son capaces de girar libremente en un sentido u otro (como un motor DC) (DIYMarkers, 2013).

Figura 9 Motor pasó a paso



Fuente: Vistronica [fotografía]. Electrónica: Motor Paso a Paso 5.6Kgcm. Disponible en <https://www.vistronica.com/robotica/motores/motor-paso-a-paso/motor-paso-a-paso-5-6kgcm-nema-17-para-impresora-3d-detail.html>

Tabla 9 Características de Motor pasó a paso

Modelo	SL42STH48-1504 ^a
Longitud	4.7cm
Angulo de paso	1.8°
Voltaje nominal	2.8V
Corriente nominal	1.68 ^a
Resistencia de fase	1.68?
Inductancia de fase	2.8mH
Torque	5.6KgrCm
Cables de conexión	4
Inercia del motor	68
Peso	350 gramos1 x Motor Paso a Paso 5.6Kgcm Nema 17 Para Impresora 3D
Costo	\$ 56.000 COP

Autor: (Vistronica, s.f.)

Tornillo sin Fin: Uno de los principales y más usados mecanismos de transmisión en cualquier proyecto mecánico es el llamado tornillo sin fin. Dicho dispositivo está formado por un sistema de un tornillo con dentado helicoidal (que actúa siempre como elemento motriz), normalmente engranado con una rueda dentada, llamada piñón o corona (que actúa como elemento conducido), de tal manera que transmite el movimiento entre ejes perpendiculares entre sí. Por cada vuelta completa del tornillo, el engranaje gira un diente, por lo que es un mecanismo capaz de ofrecer grandes reducciones de velocidad (Landin, 2018).

Características de los tornillos sin fin

- Se emplean para transmitir fuerza y movimiento entre dos ejes perpendiculares entre sí (90°), o lo que es lo mismo: transmitir un movimiento circular en el eje x al eje y.
- Es un mecanismo irreversible; es decir la corona siempre actuará como elemento conducido

- Con ellos se pueden conseguir grandes reducciones en espacios reducidos, ya que su relación de transmisión (i) es igual a $1/Z$; donde z es el número de dientes de la corona.
- Los reductores de velocidad basados en su empleo no ocupan mucho espacio. Este hecho facilita su implementación en numerosos sistemas industriales.

Figura 10 Tornillo sin fin



Fuente: Dinastía Tecnológica [fotografía]. Electrónica: Tornillo CNC / 3D / 300mm + Tuerca con Camisa 8mm. Disponible en <http://dinastiatecnologica.com/producto/tornillo-cnc-3d-300mm-tuerca-con-camisa-8mm/>

Lenguaje de Programación

Un lenguaje de programación es básicamente un sistema estructurado de comunicación, similar al humano, el cual permite comunicarse por medio de signos, ya sean palabras, sonidos o gestos. Refiriéndonos a los aparatos, este sistema está organizado para que se entiendan entre sí y a su vez interprete las instrucciones que debe ejecutar (Morales, 2014). Entre los lenguajes de programación mencionamos los más populares, estos son:

- | | | |
|----------|---------------------|---------------------|
| • Java | • C Sharp (C#) | • CSS3 |
| • C | • Visual Basic .NET | • MATLAB |
| • C++ | • JavaScript | • PL/SQL |
| • Python | • PHP | • Assembly language |
| • HTML5 | • Visual Basic | (ASL) |

En términos generales existen muchos lenguajes de programación según (Tecnología-informática, 2018) En la actualidad, **existen alrededor de 2000 lenguajes de programación**, lo que demuestra que existe un gran interés en este tipo de herramientas de diseño de software, sin embargo la mayoría de ellos son implementaciones de lenguajes más antiguos.

Cabe resaltar que a pesar de todas las posibilidades que se tiene en seleccionar un tipo de lenguaje de programación, no existe ninguno que se destaque sobre el resto, debido a que cada uno de ellos ofrece ventajas y desventajas que los hacen únicos para el tipo de desarrollo que se lleve a cabo.

Los lenguajes de programación se dividen en dos tipos: Lenguajes de Bajo Nivel y Lenguajes de alto Nivel. El primero es orientado para la máquina, en pocas palabras los que más se aproximan al “código de máquina” que son el sistema binario 0 y 1, algunos ejemplos de estos lenguajes de bajo nivel son Assembler y Lenguaje C. El segundo es orientado para resolver un problema en específico, donde el programador ya no está tan enfocado sobre las características de la computadora. Para que estos lenguajes funcionen deben contar con un traductor o compilador, el cual tienen la labor de transformar el código fuente a código objeto para ser leído por el código máquina.

A continuación se documenta algunos de los lenguajes de programación más importantes:

Lenguaje de programación C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido (Jecrespom, 2015).

Características

1. Programación de bajo nivel (bit)
2. Permite la agrupación de instrucciones
3. Posee una gran portabilidad
4. Incluye el concepto de puntero (variable que contiene la dirección de otra variable).
5. Los argumentos de las funciones se transfieren por su valor.
6. Permite la separación de un programa en módulos que admiten compilación independiente.

Lenguaje de programación PHP: es un lenguaje de código abierto, lo cual quiere decir que es de uso libre y además gratis para todos los programadores, es muy popular y se utiliza mucho para desarrollo web. Está incrustado en HTML, y esto se sabe cuándo aparece una etiqueta de la siguiente manera: `<? php` Que significa que ahí empieza el código PHP (Ramirez, 2017).

Características

1. Velocidad: PHP no solo es rápido al ser ejecutado sino que no genera retrasos en la máquina, por esto no requiere grandes recursos del sistema.
2. Estabilidad: utiliza su propio sistema de administración de recursos y posee de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
3. Seguridad: maneja distintos niveles de seguridad, estos pueden ser configurados desde el archivo **.ini**

4. Simplicidad: dispone de una amplia gama de librerías, y permite la posibilidad de agregarle extensiones. Esto le permite su aplicación en múltiples áreas, tales como encriptado, gráficos, XML y otras.

(Red Grafica Latinoamerica, 2018).

Lenguaje de programación HTML5: es la quinta revisión del lenguaje (HyperText Markup Language) HTML de marcado estándar que se emplea para la web. Es uno de los lenguajes de programación web más usados en todo el mundo, se puede crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5 (aulaformativa, 2019).

Características

1. HTML5 agrega contenido multimedia (audio y video)
2. Agrega funciones de geolocalización, animaciones, aplicaciones web, etc.
3. Es un lenguaje de programación de código abierto o libre esto quiere decir que es gratuito
4. Es un código ordenado
5. Tiene compatibilidad con los navegadores (Chrome, Firefox, Safari y Opera)
6. Mejora el almacenamiento local
7. Se puede realizar un diseño responsive (adaptable a cualquier dispositivo)

Lenguaje de programación CSS3: CSS significa Cascade Style Sheets, también llamado Hojas de Estilo en Cascada. CSS es un lenguaje de marcado que se emplea para dar formato a un sitio web. Es decir, funciona en conjunto con los archivos HTML. Sirve para cambiar el aspecto de un sitio web, desde las medidas para los márgenes hasta las

especificaciones para las imágenes y el texto. CSS3 funciona mediante módulos, algunos de los más comunes son “colors”, “fonts”, “backgrounds”, etc. Los módulos son sólo categorías en las que se pueden dividir las modificaciones que se hacen al aspecto de nuestro sitio web (Aulaformativa, 2017).

Características

1. Es gratis
2. Es fácil de aprender
3. Ofrece la posibilidad de añadir animaciones y transiciones
4. Se puede tener todos los estilos en una sola pagina
5. Genera Sitios más rápidos
6. Genera un diseño adaptable a cualquier dispositivo

Bootstrap: Es un Framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Esta técnica de diseño y desarrollo se conoce como **Responsive Design** o Diseño Adaptativo. Actualmente es uno de los Framework más usados (Caicedo , 2015).

Características

1. Utiliza componentes y servicios creados por la comunidad web, tales como:
HTML5, CSS, jQuery UI, LESS y GitHub.
2. Incluye por defecto el famoso Grid System el cual proporciona 12 columnas fijas o fluidas, dependiendo de si tu diseño será Responsive o no.

3. Soporte: Hay una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
4. Herramienta sencilla y ágil para construir sitios web e interfaces.
5. Hay mucha variedad de plantillas y temas.
6. No es ligero, y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.
7. Si se necesita añadir componentes que no existen, se debe hacer uno mismo en CSS y cuidar de que mantenga coherencia con el diseño y cuidando el responsive.

Lenguaje de programación JavaScript: Es un lenguaje de programación que permite crear contenido nuevo y dinámico, controlar archivos de multimedia, crear imágenes animadas y muchas otras cosas más. (Aunque, no todo, pero es increíble lo que se puede llegar a hacer con tan sólo unas pocas líneas de código de JavaScript) (Mozilla, 2018).

Características

1. Implementa un comportamiento dinámico
2. Permite crear una interfaz de usuario activa, lo que ofrece retroalimentación a los visitantes según navegan por sus páginas.
3. Asegura el ingreso de información válida en los formularios, lo que dará como resultado ahorro en tiempo.
4. Realiza cálculos en los formularios, sin necesidad de realizar procesos del lado del servidor.
5. Se puede crear sobre la marcha páginas HTML personalizadas, dependiendo de las acciones ejecutadas por el usuario. Suponga que está en un sitio de viajes y el usuario hace clic en Hawái como destino.

6. Controla el navegador, para que pueda abrir ventanas nuevas, visualizar ventanas de alerta e insertar mensajes personalizados en la barra de estado de la ventana del navegador.
7. Puede generar documentos con relojes calendarios y hora.
8. Puede ocuparse de formularios, establecer cookies, construir páginas HTML sobre la marcha y crear aplicaciones basadas en la Web.

(Delgado, 2019).

Lenguaje de programación Visual Basic .NET: es considerado la evolución de Visual Basic y C#, es fácil de usar y en cuanto a la hora de escribir código es el menos estricto dado que aporta la misma funcionalidad de C#, por ejemplo ambos se compilan en CIL. Además es la herramienta de desarrollo multilenguaje más completa para construir e integrar rápidamente aplicaciones y servicios Web XML. En su diseño se han integrado a fondo los estándares y protocolos de Internet, como XML y SOAP, por lo que Visual Studio .NET simplifica considerablemente el ciclo de vida del desarrollo de aplicaciones.

Características

1. Creada para los servicios Web XML
2. Constituye un modelo sencillo, flexible y basado en estándares para integrar, ampliar y publicar aplicaciones.
3. Facilita la integración en su empresa, el enlace con socios empresariales y la prestación de nuevos y eficaces servicios a sus clientes.
4. Aumenta el rendimiento, la escalabilidad y la fiabilidad

5. Elimina el problema de conflictos entre versiones. Las nuevas aplicaciones pueden ejecutar aplicaciones existentes side-byside (simultáneamente)
6. Pueden compartirse y reutilizarse las aplicaciones escritas en cualquier lenguaje de programación
7. Permite a los equipos compartir arquitecturas de aplicaciones y fomentar las mejores prácticas para rastrear rápidamente su desarrollo
8. Ofrece prestaciones de utilización “No-touch” para la instalación de aplicaciones

(Microsoft Visual Studio .net, s.f.).

Lenguaje de programación C Sharp (C#): es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework .NET. Puede usar C# para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas, muchas más cosas (Microsoft, 2015).

Características

1. La sintaxis de C# es muy similar a Java, lo que simplifica al desarrollador a la hora de escribir código.
2. Se tiene que declarar los tipos de datos con las variables, además que estas variables no pueden ser globales. De esta manera se reducen los errores de compilación.
3. Es un lenguaje orientado a objetos, así que obliga que todos los métodos y propiedades estén dentro de una clase.

4. Se pueden definir propiedades sin necesidad de crear métodos, como en Java, o usar eventos sin tratar con punteros a funciones.
5. Todos los tipos de datos sencillos de C# derivan de una clase común llamada `System.Object`. De esta manera, la conversión de tipos sólo es compatible entre los tipos más cercanos (números enteros con decimales, por ejemplo), además de facilitar la creación de colecciones.
6. Se puede recodificar un operador, como puede ser el sumatorio `+` o el incremento `++`.
7. Dispone de un recolector de basura que destruye los objetos que no se usan en memoria.
8. Puede dividir tu código en múltiples hilos de ejecución, trabajar en paralelo y sincronizándose al final.

(Lenguajes de programación, s.f.).

Lenguaje de programación MATLAB: es un lenguaje de alto desempeño diseñado para realizar cálculos técnicos. MATLAB integra el cálculo, la visualización y la programación en un ambiente fácil de utilizar donde los problemas y las soluciones se expresan en una notación matemática. MATLAB es un sistema interactivo cuyo elemento básico de datos es el arreglo que no requiere de dimensionamiento previo.

MATLAB se utiliza ampliamente en:

1. Cálculos numéricos
2. Desarrollo de algoritmos
3. Modelado, simulación y prueba de prototipos

4. Análisis de datos, exploración y visualización
5. Graficación de datos con fines científicos o de ingeniería
6. Desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI, Graphical User Interface).

La ventaja principal de MATLAB es el uso de familias de comandos de áreas específicas llamadas toolboxes. Los toolboxes son grupos de comandos de MATLAB (archivos M) que extienden el ambiente de MATLAB para resolver problemas de áreas específicas de la ciencia e ingeniería. Por ejemplo, existen toolboxes para las áreas de Procesamiento Digital de Señales, Sistemas de Control, Redes Neuronales, Lógica Difusa, Wavelets, etc. (Esqueda Elizondo, 2002).

Gestores de bases de datos

Gestor de bases de datos MySQL: MySQL “es un gestor relacional de base de datos, que organiza información en distintos archivos dependiendo el motor que se utilice y en los cuales se puede guardar un simple registro o hasta desarrollar un complejo sistema relacional orientado a objetos” (Mussa, 2016).

Es importante destacar que MySQL es uno de los gestores de bases de datos más usados. Entre los diferentes gestores de BD encontramos SQLite, Microsoft SQL Server, Oracle, Microsoft Access, entre otros. Como apunte adicional se debe tener en cuenta las ventajas y desventajas que nos proporcionan los diferentes gestores de BD ya que cada uno aporta su enfoque especial a lo que se quiere desarrollar; ítems importantes como: crear, gestionar y administrar bases de datos, así como de qué forma almacena y busca la información.

Características

1. Es una base de datos gratuita al ser de código abierto, no tiene costo.
2. Es muy fácil de usar, se puede empezar a usar la base de datos MySQL sabiendo unos pocos comandos.
3. Es una base de datos muy rápida en cuanto a rendimiento y funcionalidad.
4. Utiliza varias capas de seguridad. Contraseñas encriptadas, derechos de acceso y privilegios para los usuarios.
5. Pocos requerimientos y eficiencia de memoria. Tiene una baja fuga de memoria y necesita pocos recursos de CPU o RAM.
6. Es compatible con Linux y Windows.
7. Cuenta con soporte para aproximadamente diez motores de almacenamiento, cada uno con sus características.

Gestor de bases de datos Microsoft SQL Server: Es un sistema de gestión de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Es un sistema propietario de Microsoft. Sus principales características son:

1. Soporte de transacciones.
2. Escalabilidad, estabilidad y seguridad.
3. Soporta procedimientos almacenados.
4. Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

5. Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
6. Además permite administrar información de otros servidores de datos

Su principal desventaja es el precio, aunque cuenta con una versión EXPRESS que permite usarlo en entornos pequeños. (Aproximadamente unos 4GB de información y varios millones de registros por tabla) (Iruela, 2016).

Gestor de bases de datos PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. La comunidad PostgreSQL se denominada el PGDG (PostgreSQL Global Development Group).

Sus principales características son:

1. Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés)
2. Amplia variedad de tipos nativos: provee nativamente varios soportes
3. Ahorros considerables de costos de operación
4. Estabilidad y confiabilidad

(Iruela, 2016).

Gestor de bases de datos Microsoft Access: Es un sistema de gestión de bases de datos Relacional creado por Microsoft (DBMS) para uso personal de pequeñas organizaciones. Se ha ofrecido siempre como un componente de la suite Microsoft Office aunque no se incluye en el paquete “básico”. Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Entre las principales funcionalidades se pueden indicar las siguientes:

1. Permite crear tablas de datos indexadas.
2. Modificar tablas de datos.
3. Relaciones entre tablas (creación de bases de datos relacionales).
4. Creación de consultas y vistas.
5. Consultas referencias cruzadas.
6. Consultas de acción (INSERT, DELETE, UPDATE).
7. Formularios.
8. Informes.
9. Entorno de programación a través de VBA
10. Llamadas a la API de Windows.

(Iruela, 2016).

Gestor de bases de datos Oracle: Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Tradicionalmente Oracle ha sido el SGBS por excelencia, considerado siempre como el más completo y robusto, destacando por:

1. Soporte de transacciones.
2. Estabilidad.
3. Escalabilidad.
4. Es multiplataforma.

También siempre ha sido considerado de los más caros, por lo que no se ha estandarizado su uso como otras aplicaciones. Al igual que SQL Server, Oracle cuenta con una versión EXPRESS gratis para pequeñas instalaciones o usuarios personales (Iruela, 2016).

Servidores

Es un ordenador que permanece conectado a internet las 24 horas al día, el cual tiene como función principal proporcionar todos los contenidos necesarios para que un sitio web se visualice a un usuario a través del navegador de su preferencia. Según (Hostingsaurio, 2017) “Esto es lo que se conoce como el modelo cliente-servidor: **el cliente pide y el servidor le abastece de los recursos que necesita.** Por tanto, la información puede ser en formato texto, vídeo, audio, imágenes, emails, aplicaciones, etc.”

Existen muchos servidores donde cada uno está enfocado en una tarea específica dependiendo el requerimiento que el usuario necesite de su función. A continuación se presentan los diferentes tipos de servidores más habituales.

Servidor de correo (Mail Server): “es un ordenador que envía, recibe y almacena los mensajes de correo electrónico o emails” (Hostingsaurio, 2017). Estos se dividen en 3 servidores POP3, SMTP e IMAP.

1. POP3: Almacenan los correos entrantes lo envían a los usuarios cuando se conectan.

2. SMTP: Son los encargado de enviar los mensajes salientes
3. IMAP: Estos permiten administrar el correo electrónico sin necesidad de descargarlo en tu ordenador o en un gestor de correo.

Servidor de proxy: actúa como intermediario entre el servidor y el cliente. De este modo, el servidor no conoce la identidad del cliente. Se utiliza para mejorar la privacidad del usuario. Es habitual en los navegadores de Internet.

Servidor FTP: Es el acrónimo de File Transfer Protocol, que sería protocolo de transferencia de archivos en español. Se utiliza para enviar archivos de un ordenador a un servidor o para descargarlos desde el servidor al ordenador.

Servidor de bases de datos: este servidor almacena grandes cantidades de información, además realiza una gestión de base de datos con referente a los clientes. Para grandes cantidades de registros utiliza un clúster de servidores, el cual son un conjunto de servidores de base de datos

Servidor web: este servidor es utilizado para alojar lo sitios web de los clientes o empresa el cual es denominado Hosting. Su tarea principal consiste en almacenar todos los archivos propios de una página web (texto, imagen, vídeo, etc.) y los muestra a los clientes a través de los navegadores. Cabe resaltar que estos servidores utilizan un protocolo HTTP (Hypertext Transfer Protocol).

Los servidores web tienen una gran variedad, pero como en cada apartado se resaltan los más habituales entre ellos se tienen: Apache, Ngnix, IIS, Sun Java System y W3C Jigsaw. A continuación se realiza la definición de cada uno de ellos según (Hostingsaurio, 2017).

1. Servidor HTTP Apache: Es el servidor web más popular y uno de los más utilizados.
Es un software de código abierto que se puede instalar en casi todos los sistemas operativos, Linux, Windows e iOS entre ellos.
2. Servidor Ngnix: Es un servidor web muy ligero que utiliza el proxy inverso, es decir, que protege la identidad de los servidores, lo que mejora la seguridad de toda la información albergada en ellos.
3. Servidor Microsoft IIS: Es un servidor web desarrollado por Microsoft que ejecuta Windows a través de la tecnología IIS (Internet Information Services). Su gran ventaja es que es compatible con todas las tecnologías de la compañía.
4. Sun Java System Web Server: Es un servidor web creado para soportar una gran carga de trabajo con tecnologías muy específicas como Java, Perl, Python o Ruby on Rails. Está especialmente pensada para programadores.
5. W3C Jigsaw: Nació en el World Wide Web Consortium (W3C), un consorcio internacional para establecer los estándares de Internet. Es de código abierto y ha sido programado a partir de Java.

De acuerdo con los razonamientos que se han venido realizando es necesario comprender que se realizara la contratación de un servidor para situar el sistema en la web. Entre ellos se tienen dos opciones muy favorables el cual son los servidores VPS o Servidor Virtual Privado y los servidores Dedicados.

El servidor VPS “a aquel en el que varios usuarios comparten una misma máquina; pero, a diferencia del Alojamiento Compartido, cada uno de ellos tiene una serie de recursos garantizados, asegurando así una estabilidad.” (Axarnet, 2018)

El servidor dedicado “es la opción más exclusiva de todas ya que un usuario tiene una máquina física dedicada para él, sin tener que compartirla con ningún otro cliente.” (Axarnet, 2018) A continuación, se relaciona las diferencias que se encuentran entre estos dos Servidores:

Tabla 10 Diferencias entre servidor Virtual Privado y Servidor Dedicado

Servidor VPS o Servidor Virtual Privado	Servidor Dedicado
No tienen el control total de los servicios y configuraciones del servidor, ya que es compartido con otros usuarios.	El usuario en cuestión tiene un control pleno y absoluto sobre la configuración del propio servidor ya que es el único usuario del sistema.
Tiene un bajo rendimiento en cuanto nivel técnico de recursos ya que	Ofrece mayor rendimiento que un VPS ya que no debe compartir los recursos del hardware con otros clientes.
Su costo es relativamente bajo	Son muy costosos
Es escalable, de modo que cada usuario puede ir ampliando los recursos a medida que vaya creciendo el proyecto	No es escalable, ya que el servidor viene con todos los servicios y configuraciones para un usuario y no necesita ampliar recursos.
Cada usuario tiene garantizada la disponibilidad de los recursos; de esta manera, se evita que algunos usuarios saturen el servidor y el resto se queden sin recursos.	La disponibilidad de los recursos es plena ya que el usuario tiene una máquina dedicada enteramente a él.

Autor: Hector Fabio Duran tomando información de (Axarnet, 2018)

2.2 Marco conceptual.

De acuerdo con los razonamientos que se han venido realizando se entiende que desde un punto de vista técnico, el proceso de secado al sol para el café trae consigo muchas prácticas inadecuadas el cual hacen que el producto final no sea el más óptimo para su comercialización, y de la misma manera no sea el resultado esperado por el caficultor. Hay muchos factores negativos que intervienen en el proceso de secado del café, como puede ser una mala estructura del secadero (donde se filtre el aire), cambios climáticos, un control inadecuado sobre el tiempo para revolver el café, maltrato espontaneo al grano de café, entre otras.

Es importante destacar que durante el proceso de secado se pueden registrar altas temperaturas de acaloramiento superando los 50 °C, donde el caficultor se ve involucrado ya que el proceso para revolver el café es manual y este es un factor determinante en la investigación, porque el ser humano puede sufrir grandes problemas de salud al estar tanto tiempo expuesto a estas altas temperaturas. El caficultor puede tomar medidas de prevención para protegerse como por ejemplo: cubriéndose la cabeza con un sombrero, cubriéndose los brazos, aplicándose bloqueador para el sol; pero en muchas ocasiones no se realizan este tipo de prevenciones porque en ellos existe una tradición que los limita a no tomar este tipo de decisiones, ya que en el momento no se ven afectados y llevan consigo mucho tiempo realizando este proceso de secado manualmente, pero al paso de los años se empiezan a sobresalir los problemas de salud.

Después de lo anterior expuesto se diseñó un prototipo para ayudar a tener un control y monitoreo con respecto al secado del café sin que el caficultor tuviera la necesidad de involucrarse personalmente en el proceso. Este sistema es independiente y autónomo ya que el usuario no tendrá que tomar decisiones para llevar a cabo el proceso de secado y obtener un buen producto final.

Como se ha evidenciado anteriormente para el desarrollo de este prototipo se aplicaron una serie de tecnologías que tienen como objetivo cumplir las necesidades del proyecto y llevar a cabo una serie de tareas que tiene como resultado el control y monitoreo para el secado del café. A continuación se van a exponer y definir los diferentes tipos de tecnologías con las que se desarrolló el software.

Placas de Desarrollo

Teniendo en cuenta muchas tecnologías y con unas excelentes características de rendimiento, confiabilidad, seguridad, programación, librerías, etc. En definitiva se diseñó el prototipo mediante las placas de desarrollo Arduino UNO R3 y la Ethernet Shield. ¿Por qué la Arduino UNO R3? Simplemente porque es una placa muy completa donde le hace sobre salir su bajo costo en el mercado y sobre todo porque utiliza el microcontrolador ATmega328, el cual le permite transferencia más rápida y más memoria. A demás es importante destacar que en el diseño del prototipo no es necesario tener muchos puertos digitales y esta placa cumple con los requerimientos exigidos en la investigación.

La Ethernet Shield es un complemento para la Arduino UNO R3, el cual tiene como objetivo principal lograr una conexión mediante cable de red. Esta opción es muy completa ya que el Arduino IDE trae una librería y una serie de ejemplos que se facilitan en el momento de realizar la programación para esta placa. Cabe agregar que al estar trabajando con cable de red no habrán complicaciones en la comunicación WiFi, tal cual como sucede con el ESP8266 que la comunicación es muy inestable en ciertos lapsos de tiempo.

Sensor de Temperatura y Humedad

En este apartado se ha determinado un enfoque especial sobre la familia de sensores DHT ya que cuentan con dos modelos muy utilizados, el cual son capaces de obtener simultáneamente los valores de temperatura y humedad. Ambos sensores devuelven el valor de la medición mediante una señal digital. En este sentido sus características los hacen muy parecidos, pero su hermano mayor el DHT22 resalta unas características superiores ante su hermano menor el DHT11 que lo hacen ser la opción más favorable en la investigación. Entre ellas se tienen: El

rango de medida es mayor al DHT11 en ambos valores y su precisión es mucho mayor ante la de su hermano.

Motores

En referencia a la clasificación anterior de la investigación se tomaron tres muy opcionadas clases de motores para la implementación del prototipo funcional que fueron el motorreductor con caja reductora, el motor pasó a paso y el tornillo sin fin. Tal como se ha visto el prototipo no requiere de grandes características para llevarse a cabo por eso el tornillo sin fin es junto con un motor reductor de gran capacidad es en conclusión la opción más indicada a implementarse en el prototipo. Se resalta el económico precio que tiene en el mercado, la facilidad para su utilización y porque es compacto en su tamaño.

Lenguajes de Programación

Después de lo anterior expuesto se puede analizar que los lenguajes de programación que más se ajustan a los requerimientos para el desarrollo del prototipo funcional de control y monitoreo del secado al sol para el café son: Lenguaje C++ (Aplicado en el IDE de Arduino), PHP, Bootstrap 4 (Aplica HTML5 y CSS3) y JavaScript.

Gestores de bases de datos

El gestor de BD es muy importante ya que este tiene como funcionalidad principal almacenar los registros que se envían desde Arduino. Se requieren características esenciales para su correcto funcionamiento dentro del sistema, para ello la mejor opción y por conocimiento propio es el gestor de base de datos MySQL, debido que su interfaz es sencilla de trabajar se tiene conocimiento propio en el manejo de las sentencias SQL, bases de datos y tablas. Además

que es un gestor gratuito y de código abierto, es muy rápida, se pueden utilizar la encriptación de contraseñas para los usuarios y es compatible con Linux y Windows.

Servidores

Tal como se ha relacionado la información en el tema de los servidores para este proyecto solamente se podría tener en cuenta el servidor web, es decir solamente dos opciones fundamentales como lo es el servidor VPS o el dedicado ya que se enfoca más a los requerimientos del prototipo. Por eso se buscó un servidor dependiendo los recursos económicos del estudiante, con el objetivo de proporcionar las características mínimas requeridas para situar el proyecto en la web.

En conclusión, como se ha visto en este documento se realizó una investigación exhaustiva con el objetivo de delimitar la tecnología, los recursos y requerimientos más apropiados, el cual están ajustados a un presupuesto relativamente económico con respecto al desarrollo del diseño e implementación del prototipo funcional para el control y monitoreo de temperatura mediante el sistema de secado al sol para el café. Siguiendo este orden de ideas el procedimiento realizado para llegar a la solución de este prototipo es la siguiente:

1. En el entorno IDE de Arduino se realizó la programación adecuada para leer la información recolectada por el sensor de temperatura y humedad DHT22 en donde se establece una conexión a un archivo PHP mediante el método GET. A demás de programar cuando la temperatura excede los 50°C y la humedad sobrepasa el 20%, con el objetivo tener un control sobre los valores obtenidos en el prototipo. Debido a esto se realiza un procedimiento de control de temperatura cuando se elevan los grados centígrados, en ese preciso momento se activa una alarma y también un

ventilador para bajar la presión que está dentro del secadero solar además de realizar la acción de revolver el producto periódicamente cada cierto tiempo.

2. El archivo de programación PHP tiene como objetivo recibir los datos enviados desde el Arduino e inmediatamente insertar estos datos en una Base de Datos MySQL mediante comandos SQL.
3. Los datos almacenados en la Base de Datos son extraídos mediante un archivo PHP para poder visualizar la información recolectada de una forma amigable con el usuario.
4. El usuario podrá obtener la información mediante gráficas estadísticas y reportes finales sobre el proceso que se lleva a cabo para el secado del café.
5. El usuario no interviene en el proceso ya que es un sistema automatizado y controlado desde la programación interna del software.

Capítulo 3. Análisis del sistema actual

3.1 Identificación de las necesidades

Desde experiencias vividas y también mediante la observación sobre el proceso de secado manual para el café realizado por el caficultor, se puede decir que no se tiene un control adecuado en el tratamiento que el grano recibe, un tiempo predeterminado para revolver el café, altas temperaturas, entre otros. Para el caficultor resulta muy difícil tener un control sobre las temperaturas que se manejan en el secadero de café, además al estar expuesto a estas temperaturas podría ocasionar problemas de salud. Todos estos factores son eventualidades que se presentan en el proceso de secado del café y traería consigo irregularidades en cuanto a su producto final, ya que este proceso está ligado directamente con el caficultor y el resultado final de su producto solamente depende de él.

De acuerdo a las consideraciones anteriores se realizó el diseño de un prototipo para el monitoreo y control de temperatura mediante el sistema de secado al sol para el café, el cual tiene como objetivo principal cubrir todas las condiciones que hacen el proceso de secado manual irregular. Cabe destacar que este prototipo es automatizado y no requiere de la intervención del usuario para llevar a cabo el producto en su punto final.

3.2 Análisis Económico y Técnico

En este apartado se presenta el costo total del prototipo funcional implementado. En secciones anteriores de la investigación se ha podido identificar elementos con las características necesarias para el proyecto y de la misma manera fueron seleccionados por sus costos bajos en comparación con otros dispositivos.

Sin embargo, se resalta que el bajo costo de dicho dispositivo no debe perjudicar las características mínimas que se requieren para llevar a cabo el diseño e implementación del prototipo. Y de la misma manera que los valores presentados a continuación pueden variar dependiendo el proveedor y las promociones que este registre. A demás no se introducen los gastos de envío ya que alteran el valor real de los dispositivos tecnológicos.

Tabla 11 Costo de los elementos tecnológicos

Elemento	Precio	Cantidad	Precio Total (\$)
Arduino UNO R3	\$ 25.000	2	\$ 50.000
Ethernet Shield W5100	\$ 25.000	1	\$ 25.000
Sensor DHT22	\$ 15.000	1	\$ 15.000
Chumacera con rodamiento	\$ 8.400	2	\$ 16.800
Puente H L298n	\$ 10.000	1	\$ 10.000
Protoboard	\$ 9.000	1	\$ 9.000
Acople Flexible	\$ 15.000	1	\$ 15.000
Cable de Red	\$ 7.000	1	\$ 7.000
Tornillo CNC	\$ 45.000	1	\$ 45.000
Motor Reductor	\$ 40.000	1	\$ 40.000
DIP Switch de 8P Rojo	\$ 1.470	1	\$ 1.470
Jumpers	\$ 150	20	\$ 3.000
RTC DS3231 + Pila	\$ 12.000	1	\$ 12.000
Ventilador Mediano 12V DC	\$ 7.000	1	\$ 7.000
Módulo Relé de 2 canales	\$ 7.000	1	\$ 7.000
Conversor I2C PFC8574AT Adaptador LCD	\$ 6.000	1	\$ 6.000
LCD 16x2 Back Light Azul	\$ 10.000	1	\$ 10.000
Zumbador Buzzer Piezoelectrico 86 Db	\$ 7.000	1	\$ 7.000
Total			\$ 286.270

Autor: Hector Fabio Duran Montes

Los precios anteriormente expuestos son obtenidos de tiendas virtuales colombianas como www.vistronica.com, www.mactronica.com.co, www.dinastiatecnologica.com, entre otras

teniendo en cuenta los precios de referencia que allí se manejan. Es importante recalcar que en el proyecto se utilizan recursos de hardware globales tal como se visualiza en el siguiente cuadro.

Tabla 12 Costo de los Recursos de Hardware

Elemento	Precio total (\$)
Computador (Personal o Escritorio)	\$ 600.000
Prototipo elementos Tecnológicos	\$ 286.270
Prototipo Diseño Físico (MDF)	\$ 50.000
Total	\$ 936.270

Autor: Hector Fabio Duran Montes

Ahora se visualizan los recursos utilizados para el software del prototipo de monitoreo y control de temperatura mediante el sistema de secado al sol para el café.

Tabla 13 Costo total de los Recursos del Software

Elemento	Precio Total (\$)
Lenguaje C (Aplicado en el IDE de Arduino)	\$ 0
Lenguaje de Programación PHP	\$ 0
Bootstrap 4 (Aplica HTML y CSS3)	\$ 0
Heightcharts	\$ 0
Morris JS	\$ 0
Admin LTE	\$ 0
Total	\$ 0

Autor: Hector Fabio Duran Montes

Representación de los costos en cuanto al trabajo elaborado por las personas encargadas del prototipo.

Tabla 14 Costo total de los recursos humanos

Personal	Horas	Precio total hora (\$)	Total (\$)
Estudiante	300	\$ 25.000	\$ 7.500.000
Profesor	50	\$ 50.000	\$ 2.500.000
Total			\$ 10.000.000

Autor: Hector Fabio Duran Montes

Para concluir reunimos cada uno de los tópicos y agrupamos los costos totales para realizar un precio final sobre la elaboración del prototipo.

Tabla 15 Costo total del prototipo

Recursos	Precio total (\$)
Hardware	\$ 936.270
Software	\$ 0
Humanos	\$ 10.000.000
Total	\$ 10.936.270

Autor: Hector Fabio Duran Montes

En este orden de ideas los beneficios generados para este proyecto están enfocados en lo académico, profesional y enriquecimiento del conocimiento ya que es un prototipo a pequeña escala y por ende no se tiene un beneficio de comercialización debido a que es una primera versión Beta y se tienen muchos aspectos en los cuales pueden mejorar la tecnología a emplear.

Análisis técnico

De acuerdo con los razonamientos que se han venido realizando el sistema contara con una serie de principios básicos, el cual tendrán como objetivo el correcto funcionamiento del aplicativo. Entre ellos se están:

- Sistema Operativo Windows 7, 8, 10 (32 o 64 bits)
- Procesador Intel Inside
- Memoria RAM 2 GB
- Puertos USB
- Conexión a Internet
- Programas Instalados (XAMPP, IDE Arduino, SublimeText 3 (Puede ser cualquier otro editor de código).

Posteriormente visualizamos las diferentes vulnerabilidades, amenazas, riesgos y agrupados por categorías (hardware, software, comunicaciones, seguridad lógica, bases de datos, sistemas operativos, entre otros) que el aplicativo pueda obtener en el transcurso de su proceso automatizado.

Tabla 16 Vulnerabilidades, Amenazas, riesgos y categorías.

Nº	Vulnerabilidad	Amenazas	Riesgos	Categoría
1	Sistema operativo no Licenciado	Difusión de software Dañino	Falta de actualizar el software para poder tener una mejor utilidad del SO	Software
2	No existen planes para recuperación de información.	Infiltración de los datos principales de la empresa.	Pérdida de Información.	Software
3	Fuga de Información	Mal uso de la información repesada en el Aplicativo	Intercepción de la información	Seguridad Lógica
4	Robo de Información	Falta de control de acceso en internet	Intercepción de la información	Seguridad Lógica
5	Fallas en el sistema eléctrico y cableado.	Bajones de Energía y cableado en mal estado.	No se tiene una fuente alimentación para el computador y el prototipo y en cuestiones de un bajón de energía no soportaría dicha falla.	Hardware
6	Fácil acceso a las Bases de datos de la empresa.	Ataques cibernéticos por parte de Hackers.	Perdida de la Información o en el peor de los casos, problemas con la justicia.	Bases de datos
7	No cuenta con un Firewall o cortafuegos	Accesos no Autorizados	La no aplicabilidad en Firewall – saber qué puertos se deben bloquear o permitir, la forma de interactuar con ella o es propietario de ella, quien tiene acceso a la consola de control.	Software
8	Antivirus Desactualizados y no Licenciados.	Difusión de software Dañino	El software si no se actualizar tendría un funcionamiento erróneo en los equipos, el cual los hace vulnerables a virus	Software

9	No mantenimiento preventivo para el computador donde está instalado el aplicativo	Hay	Fallas en rendimiento del computador	el	informáticos. No hay mantenimiento para el equipo de cómputo periódicamente, esto podría incurrir en que se deterioren los mismos y vaya cortando su vida de uso.	Hardware
10	Mal organización de la Información		Proceso de Información lentitud	con la	Generalmente en la información si no cuenta con un orden, fácilmente la información de la empresa se puede ser difícil de entregar y complicar el proceso ya que no se cuenta con un orden.	Sistema de Información

Autor: Hector Fabio Duran Montes

Tomando en cuenta la normatividad de la ISO 9126 y 14598 la cual nos da a conocer los requisitos necesarios para poder establecer, implementar o mejorar sistemas de gestión de la seguridad de la información. Con el propósito de obtener un software de calidad, cumpliendo a cabalidad con todos los respaldos de seguridad, riegos y mantenimiento del aplicativo.

En términos generales se realizó un plan de pruebas en donde se aplican algunas técnicas y tipos de pruebas para mitigar los resultados anteriormente mencionados.

- Pruebas de integridad a los datos y a los datos y a la Base de Datos

Se realizara un mantenimiento incremental sobre la base de datos del aplicativo, con el objetivo de presentar un servicio de calidad, integridad y fiabilidad con respecto a la información almacenada en el sistema.

También contara con niveles de acceso a la información de la base de datos para los usuarios, con el propósito de que el usuario visualice su información en tiempo real y con un cierto nivel de seguridad propia.

- Pruebas de funcionamiento

En este punto se utilizaron las pruebas de caja blanca ya que estas están dirigidas sobre las funciones internas del sistemas, desde aquí es la parte fundamental para que el software tenga un desempeño optimo y con rapidez al momento de brindar una respuesta al usuario.

- Pruebas de interfaz de usuario

Se realizan pruebas con el objetivo de evaluar el software y todos los componentes que comprometen el control y funcionamiento adecuado de del sistema. Verificando la comunicación entre el software y el usuario, la rapidez de tiempo de respuesta y si los errores se manejan correctamente.

- Pruebas de Control de Seguridad y el Acceso

Como se ha mencionado anteriormente se tendrán unos niveles de seguridad y de acceso para el Aplicativo, en donde solamente el administrador tendrá todos los privilegios con respecto a la información almacenada en la base de datos.

Los usuarios normales tienen solamente el privilegio de revisar y analizar de una forma fácil y dinámica, además de poder ver de una forma estadística los resultados obtenidos hasta ese instante.

- Pruebas de la configuración

Es donde se realizan el proceso de probar el sistema con respecto al configuraciones de otros sistemas operativos y que de una u otra manera este sea compatible con la misma y que al momento de ser utilizada por el usuario no presente fallas.

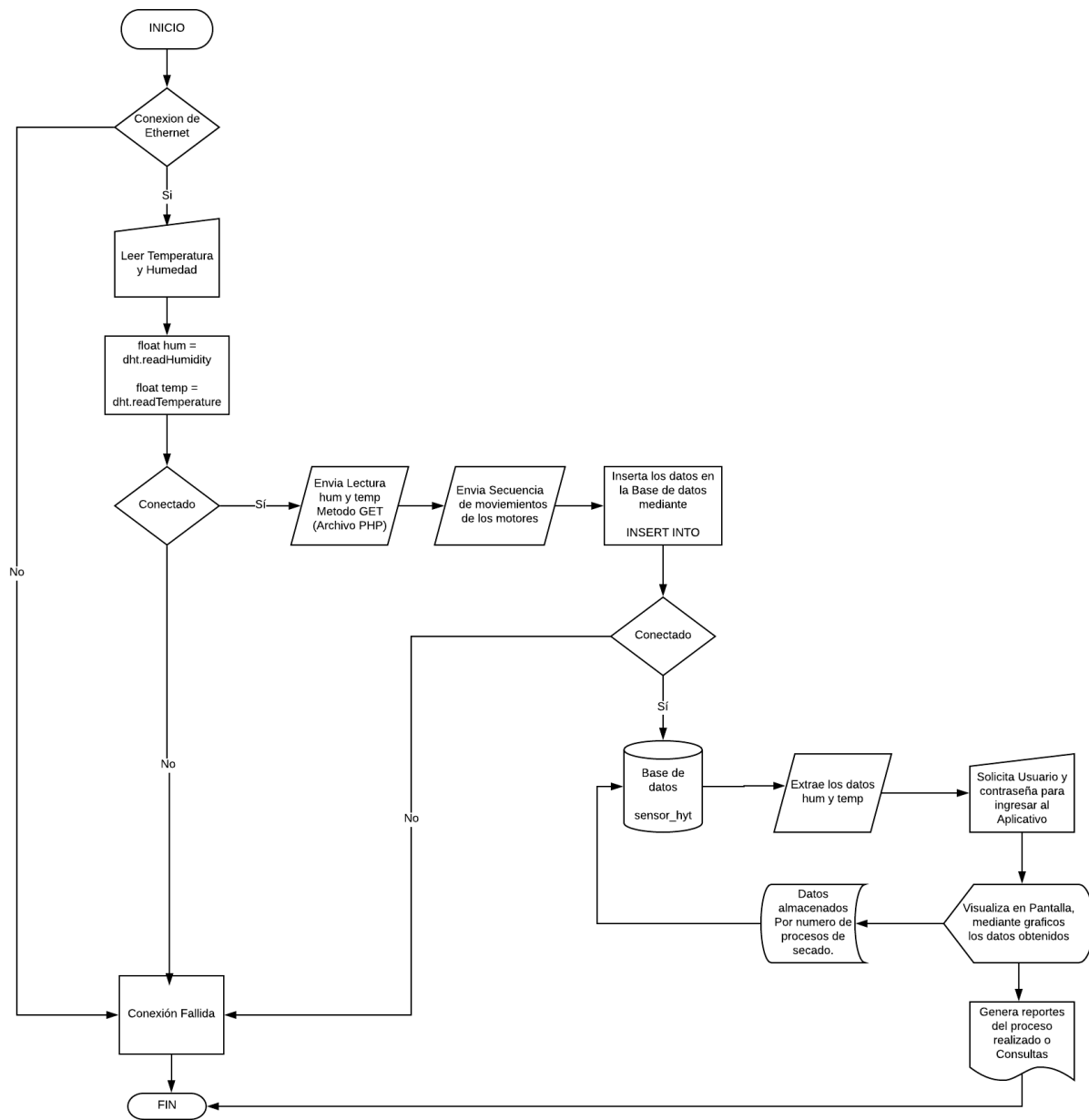
3.3 Arquitectura de sistemas (Diagrama de Flujo de Datos – Diagrama de Contexto del Sistema Actual)

Para empezar el diagrama de la figura 10 que se presenta a continuación es una representación gráfica del proceso que se lleva a cabo para obtener los datos de humedad y temperatura en tiempo real.

En primer lugar se establece la conexión de Ethernet en el IDE de Arduino mediante el lenguaje de programación C y son leídos los valores de humedad y temperatura, si esta conexión es establecida se enviarán los datos a un archivo PHP por el método GET y de igual forma es enviada la secuencia de programación de los motores. Este archivo tiene como objetivo principal insertar los datos recibidos en una Base de datos MySQL.

Posteriormente en el aplicativo el usuario digita las credenciales de autenticación para acceder a la plataforma y allí se pueden visualizar los datos extraídos de la Base de datos, el cual se realizó mediante el lenguaje de programación PHP y esta información es representada gráficamente. Cabe resaltar que el aplicativo al finalizar el proceso de secado, genera un reporte sobre la humedad y temperatura y el tiempo transcurrido durante el secamiento. Además el usuario tiene la opción de generar reportes de fechas anteriores en las que haya realizado el proceso de secado.

Figura 10. Diagrama de Flujo



Fuente: El Autor

3.4 Especificación de la arquitectura de sistemas (Explosión del DFD - Diccionario de Datos)

“El diccionario de datos es la especificación técnica necesaria para una mejor comprensión de los datos y su correcta estructuración en la fase de diseño” (sistemasumma, 2011). A continuación se describen las características fundamentales que componen un diccionario de datos.

- Nombre: Es el nombre completo del dato
- Descripción: Es una sentencia clara y precisa del origen y función del dato.
- Tipo de dato: Clasificación del tipo de información que se almacena, basándose bien en los tipos de datos primarios o en los tipos de datos de la base de datos a usar, por ejemplo: entero, flotante, fecha, hora, texto, lógico.
- Longitud: Espacio que se reservara para almacenar la información, esto se refiere al número de dígitos o caracteres que se almacenaran.
- Nulo: Especificación para indicar si este dato puede quedar vacío de forma temporal.
- Alias: Como los nombres de los datos son largos es difícil usarlos en la programación, así que se escoge un acrónimo o nombre más corto para hacer referencia a él, por ejemplo: para “Nombre del cliente” se puede usar el alias “NC”

En consecuencia a estas características y de acuerdo al diagrama de flujo presentado anteriormente se realiza la explosión del DFD – Diccionario de datos, el cual es reflejada en la siguiente tabla.

Tabla 17 Diccionario de datos – Base de datos control_secadero - tabla sensor_hyt

Nombre	Descripción	Tipo de Dato	Longitud	Nulo	Alias
Id	Llave primaria y tiene auto incremento cada vez que se sube un dato	Entero	4	No	id
Temperatura	Contiene los datos obtenidos por el sensor DHT22	Flotante	4	No	temperatura
Humedad	Contiene los datos obtenidos por el sensor DHT22	Flotante	4	No	humedad
Fecha	Almacena una fecha y hora UTC. El rango de valores oscila entre '1970-01-01 00:00:01' y '2038-01-19 03:14:07'	Fecha	10	No	fecha_hora

Autor: Hector Fabio Duran Montes

Tabla 18 Diccionario de datos – Base de datos control_secadero - tabla usuarios

Nombre	Descripción	Tipo de Dato	Longitud	Nulo	Alias
id	Llave primaria y tiene auto incremento cada vez que se sube un dato	Entero	4	No	id
Nombre	Almacena los nombre registrados en el aplicativo	Texto	Sin limite	No	nombre
Usuario	Almacena los Nick name de los usuario	Texto	10	No	usuario
Contraseña	Almacena las contraseñas registradas en el	Entero y texto	10	No	password

Perfil	aplicativo Hace referencia al perfil que desempeñara el usuario en el sistema	Texto	13	perfil
Foto	Almacena la foto que el usuario suba al aplicativo	No	No	foto
Estado	Representa si el usuario está activado o desactivado	Entero	2	estado
Ultimo ingreso	Almacena la última fecha cuando el usuario ingresa al sistema	Fecha	19	ultimo_login
Fecha	Almacena una fecha y hora UTC. El rango de valores oscila entre '1970-01-01 00:00:01' y '2038-01-19 03:14:07'	Fecha	19	fecha

Autor: Hector Fabio Duran Montes

Tabla 19 Diccionario de datos – Base de datos control_secadero - tabla clientes

Nombre	Descripción	Tipo de Dato	Longitud	Nulo	Alias
id	Llave primaria y tiene auto incremento cada vez que se sube un dato	Entero	4	No	id
Nombre	Almacena el nombre del cliente	Texto	Sin limite	No	nombre
Documento	Registra el número de identificación del cliente	Entero	Sin limite	No	documento
Correo	Almacena	Texto	Sin limite	No	email

electrónico	correo electrónico del cliente				
Teléfono	Registra el número de teléfono del cliente	Entero	10	No	telefono
Dirección	Almacena la dirección del cliente	Texto	Sin limite	No	direccion
Fecha de nacimiento	Registra la fecha de nacimiento del cliente	Fecha	10	No	fecha_nacimiento
Fecha	Almacena una fecha y hora UTC. El rango de valores oscila entre '1970-01-01 00:00:01' y '2038-01-19 03:14:07'	Fecha	10	No	fecha

Autor: Hector Fabio Duran Montes

Capítulo 4. Requerimientos del Sistema Propuesto

4.1 Creación de prototipos

El prototipo es un modelo de prueba piloto ya que se realizó con el objetivo de ponerlo en práctica y tendiente a considerar las posibilidades de un desarrollo posterior en ambientes más grandes de tamaño y con producciones de café de gran volumen. En esta ocasión el prototipo es considerado a gran escala por el desarrollo tecnológico, la facilidad de su implementación, los costos bajos que se aplicaron, el beneficio que le puede brindar a los caficultores en un futuro para el control en el proceso de secado y las utilidades que puede generar este proyecto.

Es importante destacar que se realizó un prototipo y no un sistema aplicado en ambientes reales tales como el secadero solar para el café, con el objetivo de evitar límites de efectos negativos, grandes pérdidas económicas, recursos, tiempo, etc. De esta manera ante la situación planteada el prototipo está sujeto en la obtención de pruebas y es relacionado con respecto a consecuencias positivas o negativas para proseguir en un futuro con el desarrollo del proyecto o de lo contrario será detenido.

4.2 Requerimientos de entrada/salida

Como se ha aclarado en capítulos anteriores el aplicativo contara con un servicio de seguridad media, para la cual el usuario tiene que digitar los datos de acceso al sistema muy conocidos como: usuario y contraseña, en donde el usuario es único y no podrá repetirse con respecto a los que se encuentran registrados en la Base de datos y la contraseña es de nivel de seguridad media. A continuación se exponen los niveles de contraseñas.

Según (Nadeau, 2018) Zendesk proporciona los siguientes niveles de seguridad de contraseñas.

Baja: cada contraseña debe tener un mínimo de 5 caracteres. Este es el nivel de seguridad predeterminado.

Media: cada contraseña debe tener un mínimo de 6 caracteres y cumplir con los siguientes requisitos:

- Incluir números y letras en mayúsculas y minúsculas
- Incluir un carácter especial que no sea una letra o un número

Alta: cada contraseña debe tener un mínimo de 6 caracteres y cumplir con los siguientes requisitos:

- Incluir números y letras en mayúsculas y minúsculas
- Incluir un carácter especial que no sea una letra o un número
- La contraseña vence después de 90 días y la nueva contraseña debe ser diferente a las 5 contraseñas anteriores

Personalizada (Professional y Enterprise): cada contraseña debe cumplir los requisitos que usted ha establecido. Entre las opciones, puede establecer el periodo antes de que venza la contraseña. Este nivel de seguridad solo está a disposición de los agentes y administradores.

Por otra parte el sistema contara con unos roles de usuario los cuales serán: Administrador, Especial y Cliente. Para cada uno de ellos se le presentara una salida de información independiente de acuerdo al rol que asumirá en el sistema. A continuación se presenta detalladamente la visualización de información en cada rol.

1. Administrador: Tendrá el control de todo el sistema, en donde podrá realizar modificaciones en lo que se desee. Visualizara las pestañas de Inicio, Noticias,

Monitoreo tiempo real, Gestor de usuarios, Gestor de clientes e Informes en el cual dependen los ítems Estadísticas y Reportes.

2. Especial: Este usuario tendrá la visualización de la pestaña de Inicio, Noticias y Gestor de clientes.
3. Cliente: En este rol solamente se presentara la información de salida para el usuario con respecto a su prototipo. Podrá visualizar la pestaña de Inicio, Noticias, Monitoreo en tiempo real e Informes del cual dependen los ítems Estadísticas y Reportes.

4.3 Requisitos de hardware/software

Hardware

- Computador Portátil o Escritorio
- Procesador Intel Inside, Pentium, entre otros.
- Memoria RAM 2GB
- Disco Duro 320GB
- Puertos USB
- Unidad de CD-ROM
- Placa Arduino UNO R3
- Sensor DHT22
- Ethernet Shield W5100
- Motorreductor
- Ventilador
- Puente H L9110

- Protoboard
- Batería 9V (Opcional)
- Cable de Red
- Jumpers
- Fuente de alimentación externa para Protoboard

Software

- Sistema Operativo Windows 7, 8, 10 (32 o 64 bits)
- Conexión a Internet (Broadcom)
- Programas Instalados (XAMPP, IDE Arduino, SublimeText 3 (Puede ser cualquier otro editor de código)).
- Herramientas ofimáticas
- Antivirus

Capítulo 5. Diseño de sistemas

5.1 Diseño de salida

La información que se presentara al usuario es dependiendo del rol que se registrara en el sistema, pero en general los resultados e informaciones generados son: Graficas en tiempo real, Estadísticas de la temperatura y humedad, en el cual se tendrían los siguientes apartados:

- Podrá consultar en la gráfica los valores que el usuario desee en un determinado rango de fecha, además tiene las siguientes opciones: Hoy, ayer, últimos 7 días, últimos 30 días, este mes, último mes y rango personalizado.
- Promedio diario de la temperatura y humedad
- Promedio temporada temperatura y humedad
- Punto más bajo temperatura y humedad
- Punto más alto temperatura y humedad
- Promedio más alto de temperatura y humedad por temporada
- Promedio más bajo de temperatura y humedad por temporada

En relación con este último se creara una vista la cual es llamada reportes, el cual tendrá como objetivo presentar la información en una tabla donde se agrupan los datos por fecha, registros de la temperatura y humedad de esa fecha, además el usuario tiene la posibilidad de generar estos reportes en PDF.

En la pestaña de inicio se visualizara en unas cajas de información el número de clientes y usuarios que están registrados en el sistema. Cabe agregar que también se visualizara una alerta suave para dos acciones: primero cuando el sistema detecte que ha aumentado la temperatura y

se activa el ventilador para bajar la presión en el secadero solar y segundo cuando los motores realicen el movimiento de revolver el producto del café.

5.2 Diseño de entrada

Para este propósito inicialmente se inicia hablando sobre la forma de ingreso al sistema, el cual cuenta solamente con el formulario de inicio de sesión, se preguntaran ¿porque no se realizó un formulario de registro? Sencillamente porque este sistema no está pensado para que cualquier persona logre ingresar de forma libre, sino para que una población en específico logre tener acceso a todos los servicios que ofrece esta plataforma. En este orden de ideas, solamente el administrador podrá crear usuarios, donde tendrá que completar la siguiente información en el formulario alojado en la vista de administrar usuarios.

- Nombre
- Usuario
- Contraseña
- Seleccionar perfil
- Subir foto de perfil

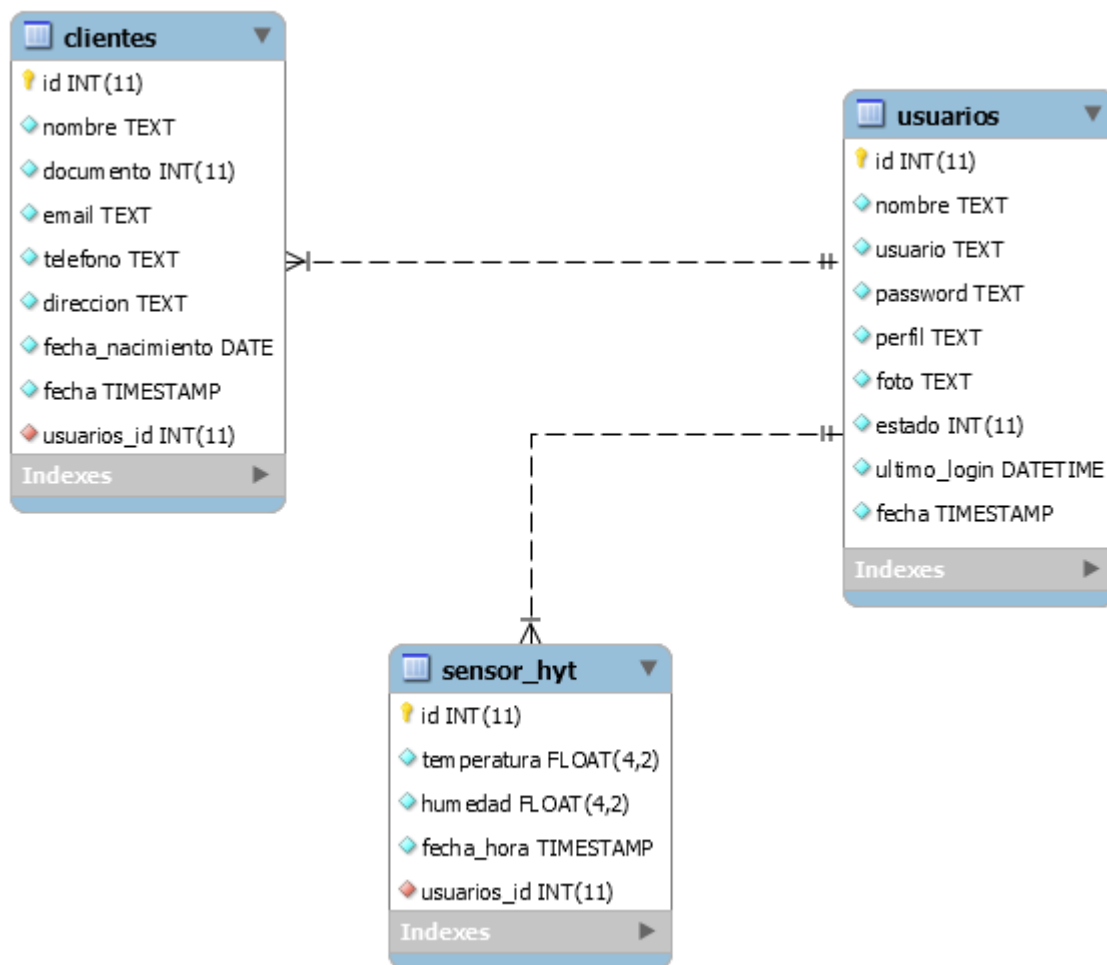
Para terminar se tiene el segundo formulario que se encuentra alojado la vista de administrar clientes, a este formulario tendrá acceso solamente el rol de administrador y el rol de Especial. Tiene como objetivo principal almacenar los datos personales de los clientes, para ello se tiene que ingresar la información correspondiente en los siguientes campos.

- Nombre
- Documento
- Correo electrónico

- Teléfono celular
- Dirección
- Fecha de nacimiento

5.3 Diseño de base de datos (Modelo Entidad/Relación)

Figura 11 Modelo Entidad Relación



Fuente: El Autor

5.4 Diseño de controles

Esta parte es fundamental para el óptimo funcionamiento de la plataforma porque los controles son los encargados de que la información que se ingrese al sistema sea válida en todo sentido y no se estén ingresando datos erróneos que puedan afectar el funcionamiento del mismo. Para ello se ha implementado controles en la vista de administración de usuarios y su respectivo formulario y también en la vista administrar clientes y su formulario de agregar clientes.

Se comienza con la administración de usuarios en donde para ingresar un usuario se debe tener en cuenta unos requisitos para su respectivo ingreso al sistema, como son:

- Nombre: No debe exceder los 30 caracteres y mínimo 4 caracteres.
- Usuario: En el sistema es identificado como Nick name, el cual representa el ingreso del usuario al sistema. Tiene como control no exceder los 15 caracteres y mínimo 4 caracteres, además que su identificación es única (esto quiere decir que no pueden haber dos usuarios con el mismo Nick name). Cuando se ha de realizar la modificación de los datos, el campo usuario no podrá ser modificado.
- Contraseña: Tiene como control no exceder los 15 caracteres y mínimo 8 caracteres. A demás cuando se registra el usuario con la contraseña que el digito, está es encriptada en la base de datos con el objetivo de prevenir futuros ataques de hackers un ejemplo seria los Ataques SQL Injection.
- Seleccionar perfil: Como estructura de control solo se podrá elegir entre los 3 perfiles que el sistema disponga, es obligatorio elegir un perfil.

- Subir foto de perfil: Este apartado cuenta con la opción de almacenar una imagen por defecto para el usuario dado en el caso que al momento de registrarse no se suba una foto personal.

Posteriormente se visualiza una alerta suave indicando que el usuario ha sido registrado con éxito y de inmediato se puede visualizar en la tabla que se encuentra en la vista de administrar usuarios. Si no se cumplen con alguno de los requerimientos antes mencionados no se lograra almacenar el usuario en la base de datos. Ahora se detallara los controles establecidos en la tabla de usuarios registrados.

En la tabla de registro se tienen 3 ítems que no se solicitan en el registro como lo son: Estado, Ultimo Login y Acciones.

- Estado: Cuando el usuario es registrado en la base de datos, este usuario inicia con el estado desactivado. El administrador tendrá el poder de activar o desactivar este usuario para poder ingresar a la plataforma.
- Ultimo Login: Es un dato de tipo fecha el cual identifica la última vez que el usuario a ingresado al sistema.
- Acciones: Este se divide en Editar información y Eliminar usuario.
 - Editar: Cuando se da clic en este botón, de inmediato se carga la información del usuario almacenada en la base de datos. En donde no se puede modificar el Nick name del usuario ya que es un identificador único, de resto cualquier campo se puede actualizar.

- Eliminar: En el momento de dar clic a este botón antes de realizar la acción de eliminar, se visualiza una alerta suave donde pregunta si ¿está seguro de eliminar el usuario? Y también tiene la posibilidad de cancelar la acción.

Seguidamente en la administración de clientes para lograr ingresar un cliente a la base de datos se tiene que cumplir con los siguientes roles en el formulario de registro.

- Nombre: No debe exceder los 30 caracteres y mínimo 4 caracteres.
- Documento: Solo permite números y no debe exceder los 15 caracteres y mínimo 5 caracteres.
- Correo electrónico: No debe exceder los 30 caracteres.
- Teléfono celular: Solo permite números y no debe exceder los 10 caracteres
- Dirección: No debe exceder los 40 caracteres y mínimo 7 caracteres.
- Fecha de nacimiento: Solo permite números y no debe exceder los 10 caracteres

Tal como se hizo en la tabla de usuarios, se visualiza una alerta suave indicando que el cliente ha sido registrado con éxito y de inmediato se puede visualizar en la tabla que se encuentra en la vista de administrar clientes. Si no se cumplen con alguno de los requerimientos antes mencionados no se lograra almacenar el cliente en la base de datos. Ahora en la tabla de clientes registrados solamente se implementan el control de acciones, el cual tienen el mismo funcionamiento que en la tabla de administración de usuarios.

Para finalizar con los controles en el la plataforma es importante destacar que en la vista de Monitoreo en tiempo real, se estableció una propiedad en la gráfica para que solamente se visualizaran 20 datos enviados desde el prototipo funcional de Arduino (Este aplica solamente en esta vista de graficas).

Por otra parte los controles no solamente están en la plataforma sino también en el funcionamiento del sistema físico del prototipo, el cual es comprendido por el entorno IDE de Arduino. Este consiste en programar los elementos electrónicos para su previo funcionamiento, en donde si la temperatura excede los 50°C se activa de inmediato un ventilador para bajar la presión del secadero solar. También existe un control de tiempo para que los motores se activen y realicen la acción de revolver el producto del café para mantener un acaloramiento equilibrado en todo el producto.

5.5 Diseño de procedimientos

Los procedimientos especifican qué tareas deben efectuarse al utilizar el sistema y quienes son los responsables de llevarlas a cabo. Procedimientos para entrada de datos, manejo de errores. En este caso la plataforma cuenta con los roles de usuarios donde los únicos que pueden efectuar tareas en el sistema son el administrados y el rol de Especial.

- **Administrador:** Su función principal es la administración de usuarios, pero también tiene acceso a toda la plataforma en general ya que su rol tiene control total de las acciones que se manejan en el sistema.
- **Especial:** Su función solamente consiste en la administración de los clientes, ya que las vistas de Inicio y Noticias no requieren de procedimientos para su funcionamiento.

Capítulo 6. Desarrollo e implementación del sistema

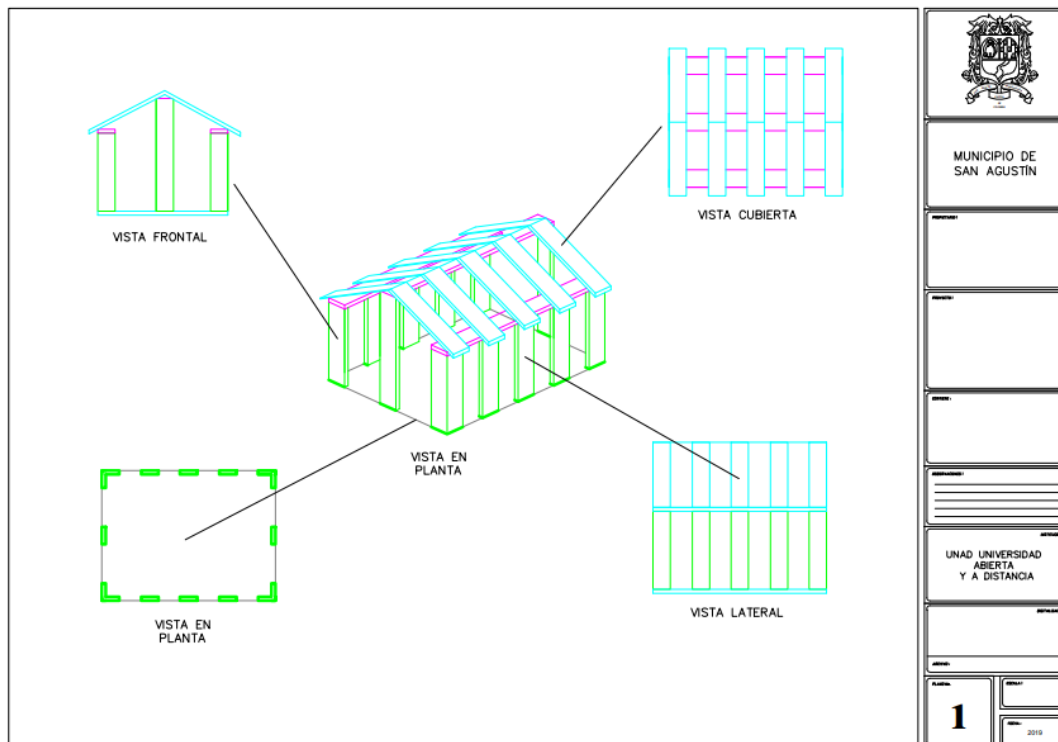
Puesta en marcha

Desarrollo del prototipo

Inicialmente se realizó un bosquejo del prototipo en AutoCAD con referente a las medidas proporcionadas para la elaboración del mismo. Estas medidas son:

- 1 Piso = 40 cm x 30 cm de 12 milímetros
- 6 Palos laterales = 8 cm x 18 cm 9 milímetros
- 1 Palo frontal = 4 cm x 26 cm 9 milímetros
- 1 Palo de atrás = 4 cm x 26 cm 9 milímetros
- 3 Palos que son las vigas = 40 cm x 4 cm de 9 milímetros
- 6 palos que son los del techo = 4 cm x 30 cm 9 milímetros

Figura 12 Bosquejo prototipo control temperatura

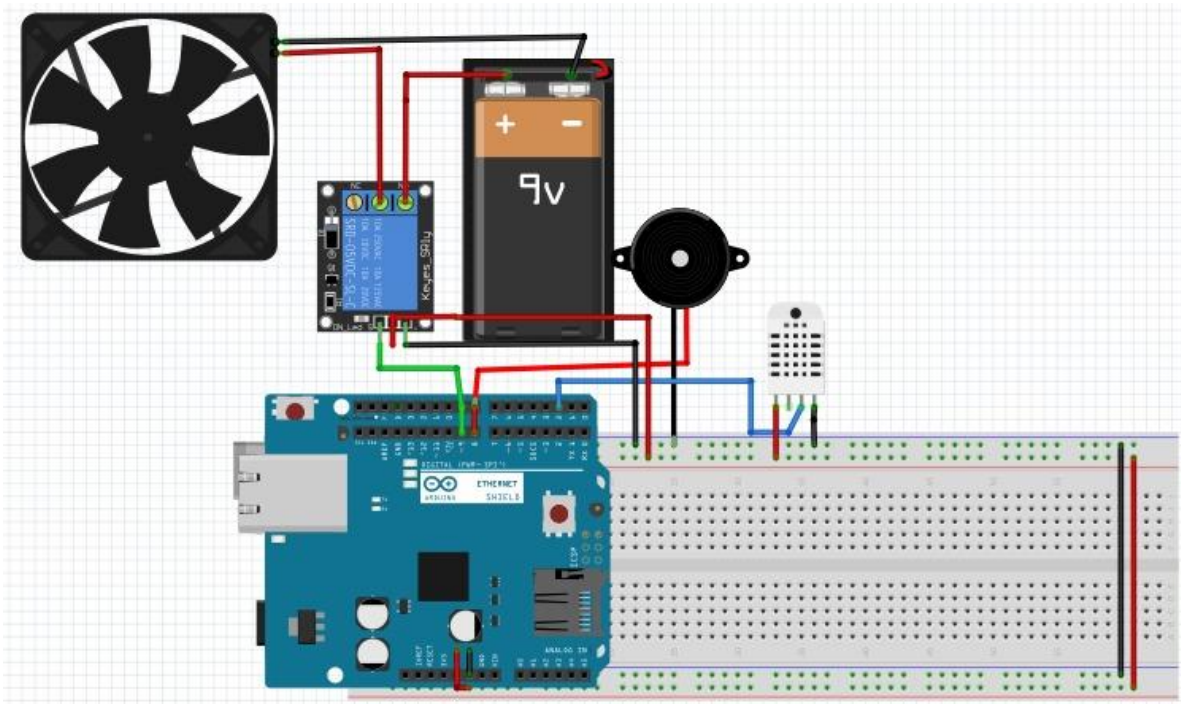


Fuente: El Autor

Cabe resaltar que el prototipo está elaborado solamente en madera MDF

Ahora se presenta a continuación las conexiones con respecto al Arduino ya que este se trabajara con dos placas un Arduino UNO. La primera placa para realizar la conexión con el sensor, el ventilador y la alarma (Buzzer) tal como se puede visualizar en la siguiente figura.

Figura 13 Prototipo - Conexión Arduino UNO – Sensor, Ventilador y Buzzer

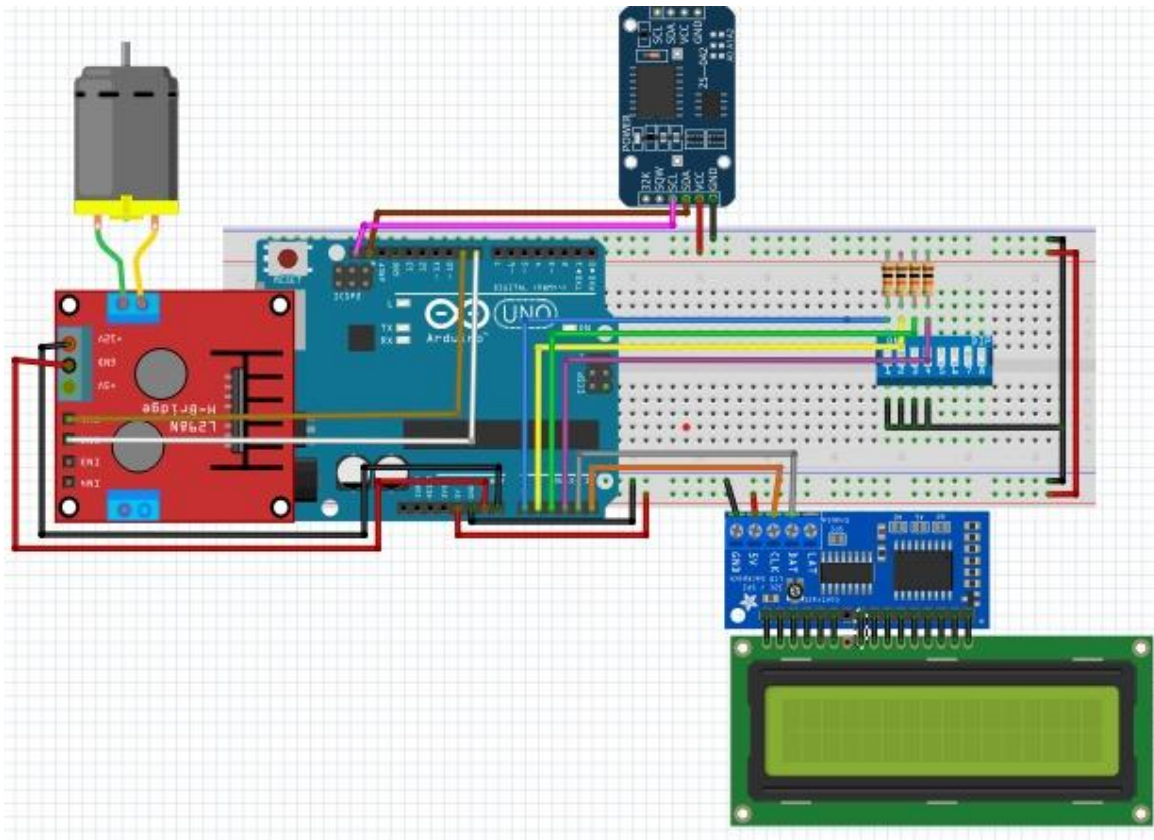


Fuente: El Autor

Para la segunda placa se realiza la conexión del motor reductor el cual tendrá como objetivo activarse en cualquier hora que se desee y a cualquier día. Esto se realiza mediante un doble Puente H L298n, un RTC de precisión DS3231 el cual servirá para activar el motor en la hora exacta, un Mini Interruptor de 8 pines para activar los días de la semana, fines de semana y festivos, también un módulo I2C para LCD, un LCD de 2x16 y por último Resistencias de 10k ohmios.

Cabe resaltar que el desarrollo de la segunda placa para activar el motor se tomó como referencia el video tutorial del Profe García “Como Automatizar Timbre de Colegio, Riego de Cultivos, Comedero de Animales”.

Figura 14 Prototipo - Conexión Arduino UNO - Motor reductor



Fuente: El Autor

Desarrollo del sistema Control Temperatura

Primero que todo es esencial especificar que este sistema está basado en el Modelo Vista Controlador explicado y desarrollado en el curso “Crea sistemas POS - Inventarios y ventas con PHP7 y AdminLTE” por Juan Fernando Urrego Instructor en Udemy y fundador de Tutoriales a tu Alcance. Se puede consultar más información en el siguiente enlace

<https://www.udemy.com/user/juanunativa/>

Hecha la observación anterior, el procedimiento que se llevó a cabo con respecto al sistema inicia con el entorno de IDE de Arduino. Como se ha mencionado en capítulos anteriores Arduino es una herramienta muy poderosa para la creación de nuevos proyectos o sistemas, el cual mejoran indispensablemente la comodidad de una persona y facilita tareas con respecto alguna actividad en específico.

Se realiza la creación de dos archivos de programación Arduino, el primero contiene la configuración del sensor DHT22 y las acciones que se realizan cuando la temperatura sobrepasa los 50°C, que en este caso es la activación de un Ventilador y un Buzzer. El segundo, tiene la configuración del movimiento que realiza el tornillo sin fin con un motor reducto, el cual tiene como objetivo principal revolver el producto del café, cada determinado tiempo para mantener una igual de acaloramiento en todo el secadero.

Ahora se realiza una explicación detallada sobre la programación de cada archivo.

En el primer documento de Arduino se encuentra inicialmente con una sección importante el cual consiste en la asignación de la dirección MAC y la dirección IP del servidor al cual se requiere la conexión.

Figura 15 Dirección MAC e IP del servidor

```
// Ingresamos la direccion MAC para el controlador
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress server(198, 54, 116, 181); //se tiene que modificar por obligacion la direccion IP de la maquina o servidor 198.54.116.181
//byte gateway ();
```

Fuente: El Autor

Seguidamente es importante destacar la asignación de unas variables con el objetivo ser llamadas más adelante para realizar la conexión con el servidor y realizar el envío de datos con una itinerancia de cada 6 minutos.

Figura 16 Asignación de variables para conexión al servidor y envío de datos

```
// Variable de tiempo de la ultima conexion en milisegundos
unsigned long ultimaConexion = 0;
// Estado de la ultima conexion
boolean ultimoEstado = false;
// Intervalo en milisegundos entre conexiones
const unsigned long intervaloConexion = 360000; //Para enviar Datos cada 6 Minutos el cual equivalen 360.000 milisegundos
```

Fuente: El Autor

Para continuar se realiza la condición en donde indica que si no está conectado, y han pasado los 6 minutos de itinerancia, entonces después de la última conexión envía los datos al servidor, en este caso de realizar en la función `httpRequest`.

Figura 17 Condición para la conexión hacia el servidor y envío de datos

```
if(!client.connected() && (millis() - ultimaConexion > intervaloConexion)) {
    httpRequest();
}
```

Fuente: El Autor

Cuando esta condición resulta verdadera entonces se realizar el procedimiento que se encuentra en la función `httpRequest`. Esta consiste en evaluar la conexión al servidor por el puerto 80 y si es verdadera, entonces se declaran las variables `temp` y `hum` para leer la temperatura y la humedad. Luego se realiza el envío de estos datos con el método GET y estos son recibidos en un archivo PHP llamado `conexión_arduino_real.php`. Finalmente se actualiza el tiempo en milisegundo con respecto a la última conexión o envío de datos exitoso, con el objetivo de ser evaluada otra vez la variable de `IntervaloConexion` y así enviar los datos cada 6 minutos.

Figura 18 Conexión Servidor

```
// Realiza la conexion http al servidor
void httpRequest() {
    // Se conecta al servidor en el puerto 80 (web)
    if (client.connect(server, 80)) {

        float hum = dht.readHumidity(); //Leemos la Humedad
        float temp = dht.readTemperature(); //Leemos la temperatura en grados Celsius

        Serial.println("conectado");
        client.print("GET /secadero/controladores/conexion_arduino_real.php?temp="); // Enviamos los datos por GET para el servidor
        client.print(temp);
        client.print("&hum=");
        client.print(hum);
        client.println(" HTTP/1.0");
        client.println("Host: hector.paginaweb.click");
        client.println("User-Agent: Arduino 1.0");
        client.println();
        Serial.println("Envio con exito!");

        // Actualiza el tiempo en milisegundos de la ultima conexion
        ultimaConexion = millis();
    }
}
```

Fuente: El Autor

Posteriormente para la inserción de datos con el método GET en la base de datos se requieren varios archivos que son:

1. conexion_arduino_real.php
2. herramienta_introducir_datos.php
3. conexion_privada.php

El primer archivo de conexión PHP requiere al segundo archivo y el segundo archivo requiere al tercer archivo. Entonces para entender el funcionamiento de estos tres archivos es preciso empezar a detallar desde el tercero hasta el primero.

conexion_privada.php: en este documento se encuentra una conexión PDO hacia la base de datos.

Figura 19 Conexión PDO a base de datos servidor

```

<?php
class conexion
{
    private $servidor;
    private $usuario;
    private $contrasena;
    private $basedatos;
    public $conexion;

    public function __construct(){
        $this->servidor = "localhost";
        $this->usuario = "hefadumo15";
        $this->contrasena = "H3f@dumo1509-";
        $this->basedatos = "control_secadero";
    }

    function conectar(){
        $this->conexion = new PDO("mysql:host=$this->servidor;dbname=$this->basedatos","$this->usuario","$this->contrasena");
    }

    function cerrar(){
        $this->conexion->close();
    }
}
?>

```

Fuente: El Autor

herramienta_introducir_datos.php: se declara una clase llamada Herramienta y dentro de ella están dos funciones principales una para concluir la conexión hacia la base de datos y la otra que es la encargada de insertar los datos temp y hum en la base de datos.

Figura 20 Inserción de datos a la base de datos del servidor

```

<?php
class Herramienta{
    public $conexion;

    function __construct(){
        require_once("conexion_privada.php");
        $this->conexion = new conexion();
        $this->conexion->conectar();
    }

    public function ingresar_datos($temp, $hum){
        $sql = " insert into sensor_hyt values (null, ?, ?, now()) ";
        $stmt = $this->conexion->conexion->prepare($sql);

        $stmt->bindValue(1, $temp);
        $stmt->bindValue(2, $hum);

        if($stmt->execute()){
            echo "Ingreso Exitoso";
        }else{
            echo "no se pudo registrar datos";
        }
    }
}
?>

```

Fuente: El Autor

conexion_arduino_real.php: En este archivo se crea la variable ingresar_dato el cual recibe los datos enviados por el archivo de Arduino con el metodo GET, segudiamente se crea una variable herramienta y se requiere este archivo para lograr insertar los datos de forma correcta en la base de datos del servidor.

Figura 21 Conexión PHP con Arduino por el método GET

```
<?php

require_once("../modelos/herramienta_introducir_datos.php");

$herramienta = new Herramienta();

$ingresar_dato = $herramienta->ingresar_datos($_GET["temp"],$_GET["hum"]);

?>
```

Fuente: El Autor

Figura 22 Verificación de datos en la base de datos del servidor

Mostrando filas 0 - 24 (total de 1859, La consulta tardó 0.0041 segundos.)

SELECT * FROM `sensor_hyt`

Número de filas: 25 Filtrar filas: Buscar en esta tabla

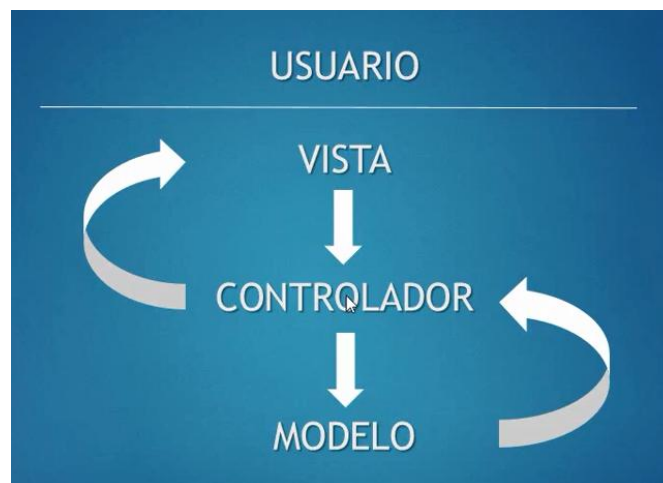
		id	temperatura	humedad	fecha_hora
<input type="checkbox"/>	Editar Copiar Borrar	3672	23.30	74.60	2019-03-03 09:36:04
<input type="checkbox"/>	Editar Copiar Borrar	3673	23.30	74.70	2019-03-03 09:44:09
<input type="checkbox"/>	Editar Copiar Borrar	3674	23.40	74.70	2019-03-03 09:50:18
<input type="checkbox"/>	Editar Copiar Borrar	3675	23.20	75.80	2019-03-03 09:56:33
<input type="checkbox"/>	Editar Copiar Borrar	3676	23.30	75.80	2019-03-03 10:02:24
<input type="checkbox"/>	Editar Copiar Borrar	3677	23.30	75.40	2019-03-03 10:08:41
<input type="checkbox"/>	Editar Copiar Borrar	3678	23.30	75.50	2019-03-03 10:14:55
<input type="checkbox"/>	Editar Copiar Borrar	3679	23.30	75.60	2019-03-03 10:20:57
<input type="checkbox"/>	Editar Copiar Borrar	3680	23.20	75.80	2019-03-03 10:27:00
<input type="checkbox"/>	Editar Copiar Borrar	3681	23.10	76.00	2019-03-03 10:33:18
<input type="checkbox"/>	Editar Copiar Borrar	3682	23.10	76.30	2019-03-03 10:39:11
<input type="checkbox"/>	Editar Copiar Borrar	3683	23.10	76.60	2019-03-03 10:45:13
<input type="checkbox"/>	Editar Copiar Borrar	3684	23.20	76.50	2019-03-03 10:51:13
<input type="checkbox"/>	Editar Copiar Borrar	3685	23.30	75.80	2019-03-03 10:57:15

Fuente: El Autor

Ahora que ya se han obtenido los datos, posteriormente se procede a la creación del sistema o sitio web que visualizara de una forma adecuada y dinámica los datos contenidos en esta base de datos donde se presentaran un monitoreo en tiempo real, graficas, reportes, etc.

Teniendo en cuenta las consideraciones anteriores es necesario un concepto sobre el patrón MVC (Models, View, Controller) o Modelo, Vista, Controlador. Es un tipo de diseño que se para en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica de la aplicación y la conexión a la base de datos, el Controlador es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas y las Vistas son las responsables de mostrar al usuario el resultado que obtiene del modelo a través del controlador. (Urrego, 2019)

Figura 23 Modelo Vista Controlador



Fuente: El Autor

Ahora después de lo anterior expuesto se indica que para el sistema se crearon una serie de carpetas con el objetivo de aplicar el modelo vista controlador que son: Controladores, Modelos y Vistas. En cada una de estas carpetas se crearon archivos que son requeridos desde el archivo de index.php tal como se evidencia a continuación.

Figura 24 Archivos requeridos en el index.php

```
<?php

require_once "controladores/clientes.controlador.php";
require_once "controladores/estadisticas.controlador.php";
require_once "controladores/noticias.controlador.php";
require_once "controladores/plantilla.controlador.php";
require_once "controladores/usuarios.controlador.php";
require_once "controladores/reportes.controlador.php";

require_once "modelos/clientes.modelo.php";
require_once "modelos/estadisticas.modelo.php";
require_once "modelos/monitoreo.modelo.php";
require_once "modelos/noticias.modelo.php";
require_once "modelos/usuarios.modelo.php";
require_once "modelos/reportes.modelo.php";

$plantilla = new ControladorPlantilla();
$plantilla -> ctrPlantilla();
```

Fuente: El Autor

Como se puede apreciar en la imagen anterior se está requiriendo el controlador plantilla.php ya que en este archivo se va a ejecutar un método que va ser instanciado desde el index.php. Por eso se crea un objeto que se llama plantilla y este objeto va instanciar una clase llamada ControladorPlantilla esta clase está creada en plantilla.controlador, por eso es que se requiere el archivo, para poder tener acceso al código que tenga ese archivo, el cual va ser el método controlador plantilla. Cuando está invocando esta clase entonces ya se puede ejecutar métodos que están dentro de esta clase en este caso ctrPlantilla, a continuación se evidencia este método que tiene como objetivo principal incluir el archivo plantilla.php el cual es el archivo principal del sistema.

Figura 25 Archivo plantilla.controlado.php

```
<?php

class ControladorPlantilla{

    static public function ctrPlantilla(){

        include "vistas/plantilla.php";

    }

}
```

Fuente: El Autor

La programación orientada a objetos es muy importante para ser trabajada en el patrón MVC a través de la instancia de clases, heredando clases y ejecutando o declarando métodos a través de objetos se puede conectar todos los estos archivos que están en diferentes carpetas.

Por las consideraciones anteriores de ahora en adelante no se entrara en gran detalle sobre la explicación que se realiza con respecto a las clases de los controladores y los métodos que allí se trabajan, solamente tendrán relevancia los métodos y clases más importantes de acuerdo a las ejecuciones que realicen en el sistema (esto es debido a la gran cantidad de líneas de código que tiene cada archivo).

De este modo se empezara a describir los archivos que componen el sistema empezando por el archivo **plantilla.php**. En este archivo se digitan las principales etiquetas de HTML5 las muy conocidas por todo programador. En la cabecera del documento se asocian los diferentes complementos que harán del sitio web más dinámico y amigable con el usuario como lo son Pugins CSS y Javascript.

En el cuerpo del documento se encuentra una instrucción de código PHP (ver figura 22) está indica que se ha establecido una variable de SESSION y recibe un parámetro el cual indica que si es correcto puede visualizar el contenido del sistema. Si genera error eso quiere decir que el usuario no está activado o registrado en la base de datos.

Figura 26 Variable de SESSION

```
if(isset($_SESSION["iniciarSesion"]) && $_SESSION["iniciarSesion"] == "ok")
```

Fuente: El Autor

Posteriormente se incluyen los archivos cabezote.php y menu.php donde en uno de estos se tienen una maquetación del sistema basado en la plantilla AdminLTE. Seguidamente se

establece un control con respecto a las rutas que se trabajaran en el sistema, entonces lo que se hace es pasar una variable GET["ruta"] con el objetivo que solamente la ruta que este en el menú logre tener acceso a la información que se presenta en el correspondiente archivo. Es decir, si se da clic en el ítem usuarios de la parte del menú, tendrá por obligación que llevarlo a la página de usuarios y si la ruta es diferente a las que están almacenadas en este archivo, entonces se visualizara una página 404 que tendrá como objetivo redireccionarlo al sistema nuevamente, el cual el usuario dará clic en un enlace para que lo posicione en la página de Inicio.

Figura 27 Control de ruta sistema

```
if (isset($_GET["ruta"])) {  
  
    if ($_GET["ruta"] == "inicio" ||  
        $_GET["ruta"] == "usuarios" ||  
        $_GET["ruta"] == "noticias" ||  
        $_GET["ruta"] == "monitoreo" ||  
        $_GET["ruta"] == "clientes" ||  
        $_GET["ruta"] == "estadisticas" ||  
        $_GET["ruta"] == "reportes" ||  
        $_GET["ruta"] == "reporteregistros" ||  
        $_GET["ruta"] == "temperaturaregistros" ||  
        $_GET["ruta"] == "maximominimoregistros" ||  
        $_GET["ruta"] == "salir") {  
  
        include "modulos/" . $_GET["ruta"] . ".php";  
    }else{  
        include "modulos/404.php";  
    }  
}
```

Fuente: El Autor

Para concluir se incluye el archivo footer.php que solamente tiene información de pie de página con respecto al sistema. Ya con esto finaliza el archivo plantilla.php y ahora se continuara con los archivos para el Login del sistema.

En archivo login.php se realiza la maquetación con respecto a un formulario el cual estará trabajando con el método post con respecto a los parámetros ingUsuario e ingPassword, en donde estos campos son requeridos obligatoriamente para ingresar al sistema. A demás al final

del formulario como se puede evidenciar se crea una clase ControladorUsuarios y un método ctrIngresoUsuario para crear el ingreso del usuario al sistema.

Figura 28 Formulario Ingreso al sistema

```
<form method="post">

  <div class="form-group has-feedback">
    <input type="text" class="form-control" placeholder="Usuario" name="ingUsuario" required>
    <span class="glyphicon glyphicon-user form-control-feedback"></span>
  </div>

  <div class="form-group has-feedback">
    <input type="password" class="form-control" placeholder="Contraseña" name="ingPassword" required>
    <span class="glyphicon glyphicon-lock form-control-feedback"></span>
  </div>

  <div class="row">
    <div class="col-xs-4">
      <button type="submit" class="btn btn-primary btn-block btn-flat">Ingresar</button>
    </div>
  </div>

<?php
  $login = new ControladorUsuarios();
  $login -> ctrIngresoUsuario();
?>

</form>
```

Fuente: El Autor

En la clase Controlador de usuarios y posicionados en el método ctrIngresoUsuarios, es de identificar primero que todo una condición el cual consiste en que si recibe el parámetro post ingUsuario entonces que se ejecute todo lo que este dentro de esta condición. En este caso un preg_match el cual tiene como objetivo controlar que solamente se acepten en el parámetro ingUsuario letras minúsculas, mayúsculas, números del 0 al 9 y las letras ñ minúscula y mayúscula, en cuanto a diferencia del parámetro ingPassword este acepta caracteres especiales. Seguidamente se crea la variable \$encriptar y se aplica la función crypt, que dentro de esta se pasa el método post del parámetro ingPassword, con la funcionalidad que en la base de datos no se logre visualizar la clave con la que se registró y este encriptada, esto para evitar ataques SQL.

Para continuar a la variable \$tabla se le asigna la tabla de usuarios que está en la base de datos, el cual fue creada con anterioridad y con algunos requerimientos en específico.

De esta manera a la variable \$item se le pasa el campo “usuario” y la variable \$valor se le asigna el parámetro ingUsuario que viene del formulario post. Luego se crea otra variable, el cual es llamada \$respuesta es la encargada de conectar con la clase ModeloUsuarios y el método MdlMostrarUsuarios en donde pase los valores de las variables antes mencionadas (\$tabla, \$item, \$valor). En este modelo primero requiere un archivo de conexión a la base de datos y luego solicita una consulta donde se trae consigo toda la tabla usuarios si es diferente de nullo la variable \$item.

Figura 29 Controlador Ingreso Usuario

```
static public function ctrIngresoUsuario(){  
    if(isset($_POST["ingUsuario"])){  
        if(preg_match('/^[a-zA-Z0-9ñÑ]+$/', $_POST["ingUsuario"]) &&  
            preg_match('/^[a-zA-Z0-9ñÑ%*](?!;=:.&#?<>");]+$/ ', $_POST["ingPassword"])){  
            $encriptar = crypt($_POST["ingPassword"], '$2a$07$asxx54ahjppf45sd87a5a4dDDGsystemdev$');  
            $tabla = "usuarios";  
            $item = "usuario";  
            $valor = $_POST["ingUsuario"];  
            $respuesta = ModeloUsuarios::MdlMostrarUsuarios($tabla, $item, $valor);  
        }  
    }  
}
```

Fuente: El Autor

Si la variable respuesta viene satisfactorio con un “ok” del modelo, entonces se realiza una condición, en donde lo único que hace es evaluar si el usuario registrado es igual a ingUsuario y si la contraseña es igual a la contraseña encriptada en la base de datos.

Figura 30 Comparación de la variable \$respuesta

```
if($respuesta["usuario"] == $_POST["ingUsuario"] && $respuesta["password"] == $encriptar)
```

Fuente: El Autor

Si todo es correcto entonces en la SESSION se pasa inicialmente el parámetro iniciarSesion igualándolo a un “ok” para que logre acceder al sistema y seguidamente se pasan los valores que están registrados en la tabla usuarios de la base de datos como lo son: id, nombre, usuario, foto y perfil.

Figura 31 Valores que están registrados en la tabla usuario

```
$_SESSION["iniciarSesion"] = "ok";  
$_SESSION["id"] = $respuesta["id"];  
$_SESSION["nombre"] = $respuesta["nombre"];  
$_SESSION["usuario"] = $respuesta["usuario"];  
$_SESSION["foto"] = $respuesta["foto"];  
$_SESSION["perfil"] = $respuesta["perfil"];
```

Fuente: El Autor

Para concluir la condición se genera una alerta con Bootstrap para cuando el usuario no está registrado e intenta ingresar al sistema.

Figura 32 Alerta Bootstrap

```
}else{  
    echo '<br><div class="alert alert-danger">Error al ingresar, vuelve a intentarlo</div>';  
}
```

Fuente: El Autor

Con referencia a lo anterior expuesto en los diferentes apartados ahora es conveniente aclarar que el sistema está compuesto por los siguientes ítems: Inicio, Noticias, Monitoreo en tiempo real, Usuarios, Clientes, Informes. Se empezara a detallar cada uno de estos explicando las funcionalidades principales e identificando la interactividad en la plataforma.

Para los módulos usuarios y clientes se implementó un CRUD que en informática es el acrónimo de Crear, Leer, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete) este se usa directamente para las funciones básicas de estos módulos como lo es agregar

usuario, agregar cliente, mostrar tabla usuarios, mostrar tabla cliente, editar usuario, editar cliente, eliminar usuario y eliminar cliente. Cabe resaltar que este CRUD se realizó tomando como referencia el curso “Crear sistemas POS - Inventarios y ventas con PHP7 y AdminLTE” por Juan Fernando Urrego”.

Gestor de usuarios

Primero se debe crear en la base de datos una tabla llamada usuarios y con los siguientes atributos.

Figura 33 Base de datos – Tabla Usuarios

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	nombre	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 3	usuario	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 4	password	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 5	perfil	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 6	foto	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 7	estado	int(11)			No	Ninguna		
<input type="checkbox"/> 8	ultimo_login	datetime			No	Ninguna		
<input type="checkbox"/> 9	fecha	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP

Fuente: El Autor

Sencillamente en esta pestaña se visualizan los usuarios registrados en la base de datos del sistema, en donde se muestran los campos nombre, usuario, la foto, el perfil (este según se haya seleccionado al momento de crear el usuario), el estado (en donde el administrador puede activar o desactivar cualquier usuario) y el registro de ultimo Login (el cual es la última fecha que el usuario ingreso al sistema). A demás se puede agregar un usuario, editar los datos de un usuario y eliminar un usuario.

Figura 34 Gestor de Usuarios

CTP Administrador

Administrar usuarios Inicio > Administrar usuarios

[Agregar Usuario](#)

Mostrar: 10 registros Buscar:

#	Nombre	Usuario	Foto	Perfil	Estado	Último Login	Acciones
1	Administrador	admin		Administrador	Activado	2019-03-15 15:40:50	
1	Hector Duran	hector		Especial	Activado	2019-03-13 18:19:06	
1	Yenifer camila	camila		Cliente	Activado	2019-03-13 18:28:32	
1	Diego Fernando Nava	diegonava		Cliente	Desactivado	2019-03-13 22:46:29	

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Copyright © 2019 Universidad Nacional Abierta y a Distancia - UNAD. Todos los derechos reservados

Fuente: El Autor

Ahora se detallaran los archivos de programación con respecto a este módulo.

En el archivo usuarios.php se encuentra la maquetación de la página, donde se realiza mediante etiquetas de Bootstrap como lo son content y box. El Box se compone de tres clases, box-header, box-body y el box-footer. En el box-body se crea la tabla el cual está asociada a un plugins de JavaScript llamada Datatables y en el cuerpo de la tabla se agregan las etiquetas PHP para hacer la visualización de los registros que se encuentran en la base de datos. Allí se delcaran 2 variables como nulas y se crea un controlados para mostrar los usuarios registrados.

Figura 35 Usuarios - Maquetación

```

<div class="box-body">

  <table class="table table-bordered table-striped dt-responsive tablas" width="100%">

    <thead>
      <tr>
        <th style="width:10px">#</th>
        <th>Nombre</th>
        <th>Usuario</th>
        <th>Foto</th>
        <th>Perfil</th>
        <th>Estado</th>
        <th>Último Login</th>
        <th>Acciones</th>
      </tr>
    </thead>

    <tbody>

      <?php

        $item = null;
        $valor = null;

        $usuarios = ControladorUsuarios::ctrMostrarUsuarios($item, $valor);

```

Fuente: El Autor

En el controlador lo único que se hace es recibir las variables nulas, crear una nueva variable para la asignación de la tabla correspondiente y la creación del modelo el cual realiza la consulta a la base de datos.

Figura 36 Usuarios - Controlador Mostrar Usuarios

```

/*=====
MOSTRAR USUARIO
=====*/

static public function ctrMostrarUsuarios($item, $valor){

  $tabla = "usuarios";
  $respuesta = ModeloUsuarios::MdlMostrarUsuarios($tabla, $item, $valor);

  return $respuesta;

}

```

Fuente: El Autor

En el método del modelo mdlMostrarUsuarios primero se recibe las variables nulas y la variable de la tabla, para ello se evalúa una condición donde dice que si es diferente a nulo, entonces realice la conexión a la base de datos y seleccione toda la tabla usuarios cuando el ítem sea igual a ítem y luego esto retorne al archivo de usuarios.php. Pero si no es así entonces se realiza una conexión a la base de datos y que traiga todos los valores de la tabla usuarios.

Figura 37 Usuarios - Modelo Mostrar Usuarios

```

/*=====
MOSTRAR USUARIOS
=====*/

static public function mdlMostrarUsuarios($tabla, $item, $valor){

    if($item != null){
        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$item");
        $stmt -> bindParam(":".$item, $valor, PDO::PARAM_STR);
        $stmt -> execute();
        return $stmt -> fetch();
    }else{

        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
        $stmt -> execute();
        return $stmt -> fetchAll();
    }

    $stmt -> close();
    $stmt -> null;
}
  
```

Fuente: El Autor

Posteriormente se crea el constructor foreach el cual “proporciona un modo sencillo de iterar sobre arrays. *foreach* funciona sólo sobre arrays y objetos, y emitirá un error al intentar usarlo con una variable de un tipo diferente de datos o una variable no inicializada”. (PHP.net)

En el archivo usuarios.php se crea un foreach para la variable \$usuarios el cual trae consigo la respuesta de la consulta realizada en el modelo MdlMostrarUsuarios allí se realiza un recorrido por los índices y se asigna el valor a \$valor y el puntero interno del array avanza en cada posición. Por eso para visualizar los datos del array se tiene que digitar un **echo** y en él

colocar las etiquetas HTML además de concatenar la siguiente expresión para visualizar los datos \$value["nombre del campo en la tabla usuarios"] a continuación se visualiza un ejemplo de ello en el código.

Figura 38 Usuarios - concatenar valores

```
foreach ($usuarios as $key => $value) {  
  
    echo '<tr>  
    <td>1</td>  
    <td>'.$value["nombre"].'</td>  
    <td>'.$value["usuario"].'</td>';  
  
    if($value["foto"] != ""){  
        echo '<td></td>';  
    }else{  
        echo '<td></td>';  
    }  
}
```

Fuente: El Autor

Ya con este procedimiento se puede visualizar los datos de los usuarios registrados en la base de datos, no entrare en detalle con respecto a cada línea digitada ya que en general todo es de la misma manera a diferencia de los botones agregar, editar y eliminar usuarios. Para las acciones o botones editar y agregar se realiza mediante una ventana modal el cual cada uno tiene un identificador único mediante la clase data-target.

Figura 39 Usuarios - Botón agregar usuario

```
<button class="btn btn-primary" data-toggle="modal" data-target="#modalAgregarUsuario">  
    Agregar Usuario  
</button>
```

Fuente: El Autor

La ventana modal es un complemento de Bootstrap el cual es un cuadro de dialogo o una ventana emergente que se muestra en la parte superior de la página actual. La maquetación de

este modal es sencilla ya que solamente se requiere etiquetas HTML y en ella se crea un formulario con el método post, donde se enviarán los siguientes parámetros: nuevoNombre, nuevoUsuario, nuevoPassword, nuevoPerfil y nuevaFoto. Cabe agregar que el modal se compone de tres clases modal-header (es la cabecera del modal donde se puede agregar un título), modal-body (es el cuerpo del modal, donde se crea el formulario) y modal-footer (es el pie del modal, donde se creó dos botones “salir” y “guardar usuario” este último de tipo submit).

Antes de cerrar la etiqueta del formulario se crea la clase ControladorUsuarios y el método ctrCrearUsuario este es ejecutado en los archivos de controlador y el modelo para el usuario.

Figura 40 Usuarios - Controlador Crear Usuario

```
<!--=====
PIE DEL MODAL
=====-->

<div class="modal-footer">
  <button type="button" class="btn btn-default pull-left" data-dismiss="modal">Salir</button>
  <button type="submit" class="btn btn-primary">Guardar usuario</button>
</div>

<?php
  $crearUsuario = new ControladorUsuarios();
  $crearUsuario -> ctrCrearUsuario();
?>
</form>
```

Fuente: El Autor

En este método del controlador se realiza primero la validación de los datos a ingresar en el sistema, el cual corresponde a que el campo nuevoNombre solamente acepta letras minúsculas, mayúsculas, números del 0 al 9, la ñ en minúscula y mayúscula además de las vocales con tildes. El parámetro nuevoUsuario acepta letras minúsculas, mayúsculas y números

del 0 al 9. El parámetro nuevoPassword a diferencia del usuario acepta también caracteres especiales.

Figura 41 Usuarios - Validación de la información en el controlador

```
static public function ctrCrearUsuario(){
    if (isset($_POST["nuevoUsuario"])){
        if (preg_match('/^[a-zA-Z0-9ñÑáéíóúÁÉÍÓÚ ]+$/ ', $_POST["nuevoNombre"]) &&
            preg_match('/^[a-zA-Z0-9]+$/ ', $_POST["nuevoUsuario"]) &&
            preg_match('/^[a-zA-Z0-9%*](?!;=:.&#?¿<>");]+$/ ', $_POST["nuevoPassword"])){
```

Fuente: El Autor

Posteriormente se crea la variable \$tabla en donde se le asigna la tabla de usuarios que se encuentra en la base de datos y la creación de la variable de encriptación para la contraseña. Seguidamente es creada una variable llamada \$datos y en este lo que se hace es enviar un array que almacena los parámetros del formulario post en unas propiedades como lo son: nombre, usuario, Password, perfil y foto. Finalmente se crea la clase ModeloUsuarios y el método mdlIngresarUsuarios tal como se evidencia en la figura 37.

Figura 42 Usuarios - Encriptación y creación del Modelo ingresar usuario

```
$tabla = "usuarios";
$encriptar = crypt($_POST["nuevoPassword"], '$2a$07$asxx54hjppf45sd87a5a4dDDGsystemdev$');

$datos = array("nombre" => $_POST["nuevoNombre"],
               "usuario" => $_POST["nuevoUsuario"],
               "password" => $encriptar,
               "perfil" => $_POST["nuevoPerfil"],
               "foto" => $ruta);

$respuesta = ModeloUsuarios::mdlIngresarUsuario($tabla, $datos);
```

Fuente: El Autor

En este método se reciben los dos parámetros del controlador, además se crea el objeto \$stmt para realizar la conexión a la base de datos y de la misma manera va a insertar en la tabla

los parámetros almacenados en la variable \$datos del controlador con sus respectivos VALUES para que se conviertan en parámetros. Seguidamente se enlazan los parámetros con el bindParam con respecto a los valores que vengan desde el \$datos["parámetro"] y si esto se ejecuta correctamente entonces se devuelve un "ok" y si no se devuelve un "error". Ya por último se cierra la conexión y se vacía el objeto.

Figura 43 Usuarios - Modelo Mostrar Usuarios

```
/*=====
REGISTRO USUARIOS
=====*/

static public function mdlIngresarUsuario($tabla, $datos){

    $stmt = Conexion::conectar()->prepare("INSERT INTO $tabla(nombre, usuario, password, perfil, foto) VALUES (:nombre, :usuario, :password, :perfil, :foto)");

    $stmt->bindParam(":nombre", $datos["nombre"], PDO::PARAM_STR);
    $stmt->bindParam(":usuario", $datos["usuario"], PDO::PARAM_STR);
    $stmt->bindParam(":password", $datos["password"], PDO::PARAM_STR);
    $stmt->bindParam(":perfil", $datos["perfil"], PDO::PARAM_STR);
    $stmt->bindParam(":foto", $datos["foto"], PDO::PARAM_STR);

    if($stmt->execute()){
        return "ok";
    }else{
        return "error";
    }

    $stmt->close();
    $stmt = null;
}
```

Fuente: El Autor

Finalmente en la sección modal agregar usuario si todo está en orden y se devuelve un "ok" entonces se mostrara una alerta suave (esta alerta está asociada a la plantilla y se llama SweetAlert) indicando que el usuario se ha registrado correctamente o de lo contrario informando que no se ha podido ingresar el usuario a la base de datos.

Ya con esto se concluye el modal de Agregar Usuario puesto que los controles, consultas y procedimientos del modelo vista controlador se han especificado en brevedad con respecto al código proporcionado para esta acción.

En segundo lugar tenemos la acción de Editar usuario, el cual corresponde al `modalEditarUsuario`. Tal como se evidencio en el modal agregar usuario, este es el mismo procedimiento para la maquetación a diferencia que los parámetros que se trabajaran son los siguientes: `editarNombre`, `editarUsuario`, `editarPassword`, `editarPerfil` y `editarFoto`. En cuanto al pie del modal, la diferencia está en el nombre del botón que en este caso es “Modificar Usuario” y de la misma manera antes de cerrar la etiqueta del formulario se crea en la clase `ControladorUsuarios` el método `ctrEditarUsuarios`.

Figura 44 Usuarios - Footer del modal agregar usuario

```
<!--=====
PIE DEL MODAL
=====-->

<div class="modal-footer">
  <button type="button" class="btn btn-default pull-left" data-dismiss="modal">Salir</button>
  <button type="submit" class="btn btn-primary">Modificar usuario</button>
</div>

<?php
  $editarUsuario = new ControladorUsuarios();
  $editarUsuario -> ctrEditarUsuario();
?>

</form>
```

Fuente: El Autor

En este método del controlador primero se realiza una condición donde indica que si se ha recibido un parametro “`editarUsuario`” entonces se ejecute una serie de instrucciones en esa condición, tal cual como se realizó en el modal Agregar Usuario. Posteriormente esta otra condición que es el `preg-match` para la validación de información que se ingresara en el parametro “`editarNombre`”

Figura 45 Usuarios - Validación de información Controlador Editar Usuario

```
/*=====
EDITAR USUARIO
=====*/

static public function ctrEditarUsuario(){

    if(isset($_POST["editarUsuario"])){

        if(preg_match('/^[a-zA-Z0-9ñÁéíóúÁÉÍÓÚ ]+$/', $_POST["editarNombre"])){
```

Fuente: El Autor

Ahora se identifica la creación de la variable \$tabla ya muy conocida en los proceso que se llevan a cabo en el sistema, donde se almacena la tabla usuarios que está en la base de datos. Con respecto a la edición de la contraseña se realiza una condición, la cual consiste en que si el parámetro post de editarPassword es diferente a vacío, entonces realiza la validación del parámetro editarPassword y encripta la contraseña, además si la contraseña se envía vacía se mostrara una alerta suave indicando que no puede estar vacía. Esto sucede solo cuando se modifica la contraseña, si no se modifica la contraseña entonces no hay problema esta es almacena en el parámetro passwordActual el cual está de tipo hidden en el formulario de usuarios.php.

Cuando se ha validado esta condición entonces lo que se hace declarar nuevamente la variable \$datos la cual contiene el array que almacena los parámetros del formulario post en las respectivas propiedades ya definidas anteriormente. Finalmente se crea en la clase ModeloUsuarios el método mdlEditarUsuario y se pasan los parámetros \$tabla y \$datos.

Figura 46 Usuarios - Alerta Suave

```
$tabla = "usuarios";

if($_POST["editarPassword"] != ""){

    if(preg_match('/^[a-zA-Z0-9%*]($!;=:.&#?¿<>);]+$/ ', $_POST["editarPassword"])){

        $encriptar = crypt($_POST["editarPassword"], '$2a$07$asxx54ahjppf45sd87a5a4dDDGsystemdev$');

    }else{

        echo'<script>

            swal({
                type: "error",
                title: "¡La contraseña no puede ir vacía!",
                showConfirmButton: true,
                confirmButtonText: "Cerrar"
            }).then(function(result){

                if (result.value) {
                    window.location = "usuarios";
                }
            })

        </script>';

    }

}else{

    $encriptar = $_POST["passwordActual"];

}

$datos = array("nombre" => $_POST["editarNombre"],
               "usuario" => $_POST["editarUsuario"],
               "password" => $encriptar,
               "perfil" => $_POST["editarPerfil"],
               "foto" => $ruta);

$respuesta = ModeloUsuarios::mdlEditarUsuario($tabla, $datos);
```

Fuente: El Autor

En consecuencia ahora en el método mldEditarUsuario tiene la tarea de recibir los parámetros anteriormente establecidos, va a generar una conexión a la base de datos y va ejecutar la sentencia SQL UPDATE (actualizar) en la tabla usuarios y en los campos nombre, Password, perfil y foto cuando usuario coincida con el usuario que viene del formulario post editar Usuario. Luego se enlazan los parámetros con el bindParam tal cual como se realizó en la consulta para

agregar un usuario y si esto se ejecuta correctamente entonces se devuelve un “ok” y si no se devuelve un “error”. Ya por último se cierra la conexión y se vacía el objeto.

Figura 47 Usuarios - Modelo Editar Usuario

```
static public function mdlEditarUsuario($tabla, $datos){  
    $stmt = Conexion::conectar()->prepare("UPDATE $tabla SET nombre = :nombre, password = :password, perfil = :perfil, foto = :foto WHERE usuario = :usuario  
    ");  
    $stmt -> bindParam(":nombre", $datos["nombre"], PDO::PARAM_STR);  
    $stmt -> bindParam(":password", $datos["password"], PDO::PARAM_STR);  
    $stmt -> bindParam(":perfil", $datos["perfil"], PDO::PARAM_STR);  
    $stmt -> bindParam(":foto", $datos["foto"], PDO::PARAM_STR);  
    $stmt -> bindParam(":usuario", $datos["usuario"], PDO::PARAM_STR);  
    if($stmt -> execute()){  
        return "ok";  
    }else{  
        return "error";  
    }  
    $stmt -> close();  
    $stmt = null;  
}
```

Fuente: El Autor

Finalmente en la sección modal editar usuario si todo está en orden y se devuelve un “ok” entonces se mostrara una alerta suave indicando que el usuario se ha modificado correctamente o de lo contrario informando que no se ha podido modificar el usuario a la base de datos.

Por otra parte para la acción o el botón Eliminar usuario, se crea en la clase controladorUsuarios el método ctrBorrarUsuario para finalizar las funciones principales del módulo de gestor de usuarios.

Figura 48 Usuarios - Controlador Eliminar Usuario

```
<?php  
  
    $borrarUsuario = new ControladorUsuarios();  
    $borrarUsuario -> ctrBorrarUsuario();  
  
?>
```

Fuente: El Autor

De esta manera el método para borrar usuario a diferencia de los otros métodos este te trabaja con GET y no con POST debido que en el botón se crea la clase idUsuario el cual envía por este método el id del usuario que se encuentra registrado en la base de datos (ver figura 45). Entonces en el controlador si este parámetro es recibido o identificado, se procede a realizar los procesos que se encuentran en esta condición (ver figura 46).

Figura 49 Usuarios - Botón Eliminar usuario

```
<button class="btn btn-danger btnEliminarUsuario" idUsuario="'.$value["id"].'" fotoUsuario="'.$value["foto"].'" usuario="'.$value["usuario"].'"><i class="fa fa-times"></i></button>
```

Fuente: El Autor

Figura 50 Usuarios - Condición para borrar un usuario

```
static public function ctrBorrarUsuario(){  
    if(isset($_GET["idUsuario"])){  
        $tabla ="usuarios";  
        $datos = $_GET["idUsuario"];
```

Fuente: El Autor

Como en los otros métodos aquí también se genera el modelo para realizar la consulta en la base de datos y así se pueda borrar de forma exitosa el usuario. Para ello se creó en la clase ModeloUsuarios el método mdlBorrarUsuario donde se le pasa los parámetros de \$tabla y \$datos.

Figura 51 Usuarios - Creación del Modelo Borrar Usuarios

```
$respuesta = ModeloUsuarios::mdlBorrarUsuario($tabla, $datos);
```

Fuente: El Autor

En el modelo borrar usuario simplemente se genera la conexión a la base de datos y se utiliza la sentencia SQL DELETE en donde se selecciona la tabla (usuarios) cuando el id es igual

al enviado desde la acción o el botón eliminar. Luego se enlaza el parámetro id con el bindParam y si esto se ejecuta correctamente entonces se devuelve un “ok” y si no se devuelve un “error”.

Ya por último se cierra la conexión y se vacía el objeto.

Figura 52 Usuarios - Modelo Borrar Usuario

```
static public function mdlBorrarUsuario($tabla, $datos){
    $stmt = Conexion::conectar()->prepare("DELETE FROM $tabla WHERE id = :id");
    $stmt -> bindParam(":id", $datos, PDO::PARAM_INT);

    if($stmt -> execute()){
        return "ok";
    }else{
        return "error";
    }

    $stmt -> close();
    $stmt = null;
}
```

Fuente: El Autor

Gestor de Clientes

Primero se debe crear en la base de datos la tabla clientes y con los siguientes atributos.

Figura 53 Clientes - Base de datos Tabla clientes

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	nombre	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 3	documento	int(11)			No	Ninguna		
<input type="checkbox"/> 4	email	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 5	telefono	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 6	direccion	text	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/> 7	fecha_nacimiento	date			No	Ninguna		
<input type="checkbox"/> 8	fecha	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP

Fuente: El Autor

Este módulo tendrá como objetivo registrar la información personal de los clientes en una base de datos totalmente a parte de la de usuarios ¿porque es indispensable tener esta información de esta manera? Ya que el prototipo en este momento no está funcionando si no solamente para una sola placa de Arduino se podría decir que no es necesaria, pero en el

momento en que se realice una mejora para trabajar con cientos de Arduinos en esta plataforma se tendrá que llevar un registro aparte de los clientes, con el objetivo de aplicar con más énfasis el Frontend del sistema, ya que esto va del lado del cliente como por ejemplo: Tener un calendario con las fechas de cumpleaños de los clientes y enviarles un mensaje de felicitación por el acontecimiento, quizás también enviar información sobre el manejo del cultivo al celular de forma periódica, publicidad de nuevos productos, etc. Mejor dicho se puede sacar indispensablemente el provecho a esta tabla de datos.

Figura 54 Clientes - Gestor de Clientes

#	Nombre	Documento ID	Email	Teléfono	Dirección	Fecha nacimiento	Fecha Sistema	Acciones
1	Hector Fabio Duran	1083909330	hefadumo15@live.com	(313) 210-6973	Calle 11 # 3 - 60 Cambulos	1994-09-15	2019-03-03 22:02:20	
2	Yenifer Guerrero	1082807454	kamilita_23@hotmail.com	(312) 876-1845	cr 5 # 2 - 08	1997-04-26	2019-03-03 22:05:58	

Mostrando registros del 1 al 2 de un total de 2

Anterior 1 Siguiente

Copyright © 2019 Universidad Nacional Abierta y a Distancia - UNAD. Todos los derechos reservados

Fuente: El Autor

Por otra parte como se puede evidenciar a simple vista el gestor de clientes es una copia del CRUD realizado en el gestor de usuarios, su diferencia es precisamente la tabla de clientes y los campos que ella contiene, puesto que las consultas y procedimientos realizados en los controladores y modelos son los mismos que se aplicaron en el gestor de usuarios, a diferencia que los parámetros que en esta se solicitan son con diferentes nombre. A continuación se hace mención sobre de los parámetros que contienen el archivo clientes.php.

- clientes.php
 - Modal Agregar Cliente
 - nuevoCliente
 - nuevoDocumentoId
 - nuevoEmail
 - nuevoTelefono
 - nuevaDireccion
 - nuevaFechaNacimiento
 - Modal Editar Cliente
 - editarCliente
 - editarDocumentoId
 - editarEmail
 - editarTelefono
 - editarDireccion
 - editarFechaNacimiento

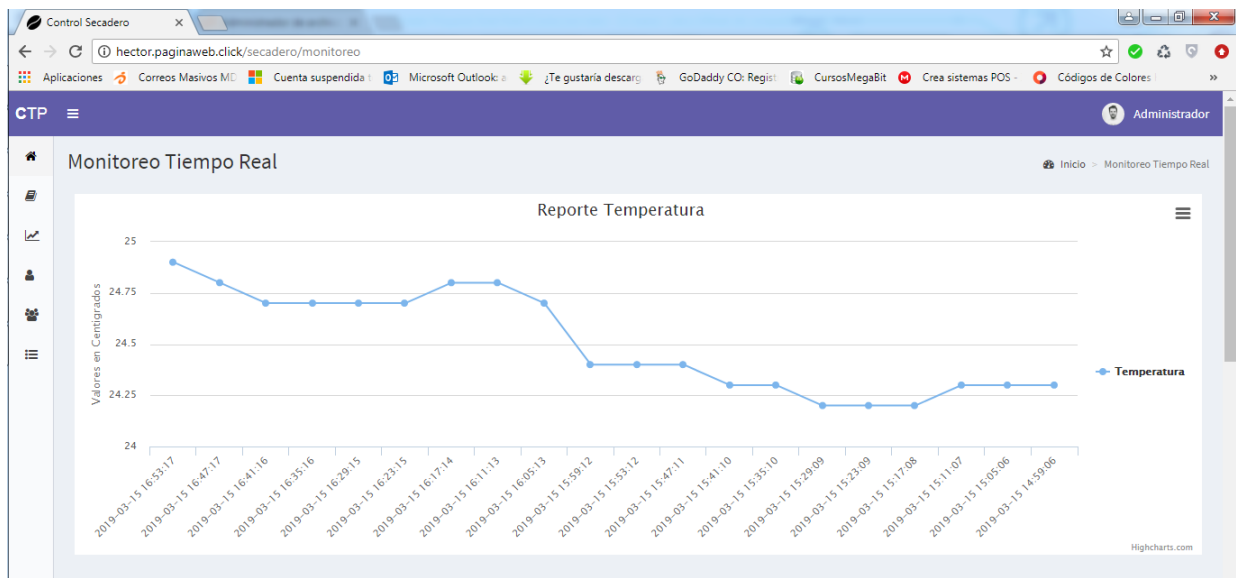
Cabe resaltar que el nombre de los controladores también todos son diferentes a los que se utilizaron en el gestor de usuarios. Entre ellos están:

- clientes.controlador.php
 - Clase ControladorClientes
 - ctrCrearCliente
 - ctrMostrarClientes
 - ctrEditarCliente
 - ctrEliminarCliente
- clientes.modelos.php: Requiere la conexión a la base de datos
 - Clase ModeloClientes
 - mdlIngresarCliente
 - mdlMostrarClientes
 - mdlEditarCliente
 - mdlEliminarCliente

Monitoreo tiempo real

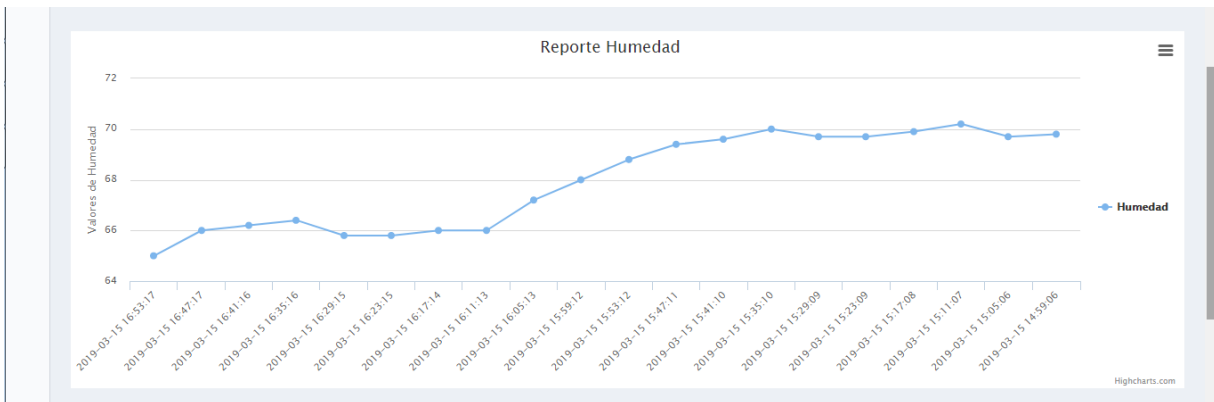
En este módulo es preciso especificar claramente que estas graficas son generadas por un complemento llamado Highcharts el cual es un frameworks que facilita el trabajo con respecto a la ilustración de la información en forma dinámica, análisis de los datos, gráficos en diferentes formatos como: barras, líneas, círculo, área, etc.

Figura 55 Monitoreo tiempo real – Grafica temperatura



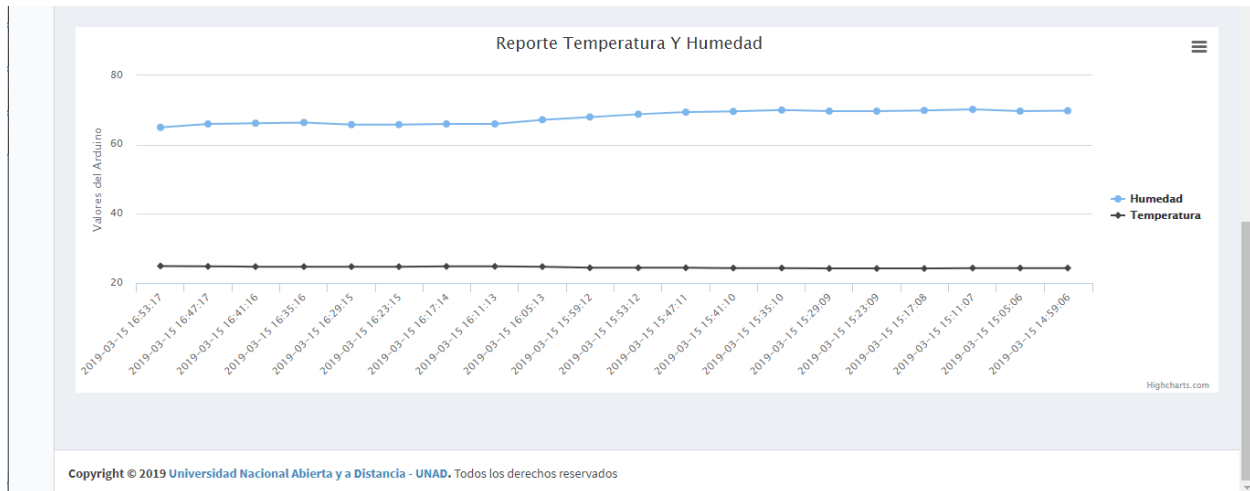
Fuente: El Autor

Figura 56 Monitoreo tiempo real – Grafica humedad



Fuente: El Autor

Figura 57 Monitoreo tiempo real – Grafica temperatura y humedad



Fuente: El Autor

Hecha la observación anterior con respecto las gráficas, ahora se sitúa en el código que proporciona Highcharts para realizar las diferentes graficas que se visualizaron en las figuras 51, 52 y 53.

Antes que nada lo primero que se debe realizar es un archivo con extensión .PHP el cual realice una conexión a la base de datos totalmente diferente a las conexiones con la que se trabajó para el login y los modelos del sistema. Este archivo tiene como nombre conexión_sensor_hyt.php el cual contiene el siguiente código de conexión.

Figura 58 Monitoreo Tiempo real - Conexión base de datos

```
<?php
$connection = mysqli_connect("localhost","hefadumo15","H3f@dumo1509-","control_secadero");
?>
```

Fuente: El Autor

Después de generar la conexión, el siguiente paso es descargar de la documentación principal de Highcharts unas librerías para el correcto funcionamiento de las gráficas, estos archivos son:

- exporting.js
- highcharts.js
- jquery.js

Por lo tanto ahora se enfocara más sobre el archivo que genera la primera grafica (ver figura 51) el cual tiene como nombre grafica_temperatura.php, en este archivo lo primero que se debe realizar abrir etiquetas PHP antes del código de la gráfica y requerir el archivo de conexión a la base de datos tal como se visualiza a continuación.

Figura 59 Monitoreo Tiempo real - Requiriendo archivo de conexión

```
<?php  
require_once("conexion_sensor_hyt.php");  
?>
```

Fuente: El Autor

En segundo lugar en el código de la gráfica se redirecciona la ruta de los archivos que fueron descargados desde la documentación de Highcharts y posteriormente se modifican todas las etiquetas textuales con referente a las necesidades que se tienen que presentar en la gráfica de temperatura. Seguidamente se implementa un código JavaScript para que la gráfica se esté recargando cada cierto tiempo en este caso cada 10.000 milisegundos.

Figura 60 Monitoreo Tiempo real - Recargo de la página

```
<script type="text/javascript">
  setTimeout('document.location.reload()',10000)
</script>
```

Fuente: El Autor

Las gráficas de Highcharts se componen de categorías (columnas) y series (filas) estas se ubican en las etiquetas **xAxis** y **yAxis** en cada una de estas etiquetas se realizara una consulta PHP para traer los registros que se encuentran en la base de datos.

xAxis: en esta etiqueta primero se crea una variable llamada \$sql y allí se realiza la consulta mediante la sentencia SQL SELECT el cual selecciona el fecha_hora de la tabla sensor_hyt ordenado por id en forma descendente y con un límite máximo de 20 valores para que se visualicen en la gráfica. Seguidamente se crea \$result el cual envía la consulta a la base de datos. Finalmente se crea una bucle while con el objetivo de obtener un array asociativo de la variable \$result y este imprima con un echo los registros que están en el campo fecha_hora.

Figura 61 Monitoreo Tiempo real - Consulta PHP para la gráfica temperatura

```
xAxis: {
  categories: [
    <?php
      $sql = " select fecha_hora from sensor_hyt order by id desc limit 20 ";
      $result = mysqli_query($connection, $sql);
      while($registros = mysqli_fetch_array($result)){
    ?>
        '<?php echo $registros["fecha_hora"]?>',
        <?php
      }
    ?>
  ]
},
```

Fuente: El Autor

yAxis: Esencialmente se enfocara en la etiqueta series ya que es en donde se realiza la consulta para visualizar los datos en la gráfica. En esta etiqueta se realiza la misma consulta que

la anterior a diferencia que en este caso solamente se imprimen los valores de temperatura y si todo sale correctamente se visualizarán los datos en la gráfica temperatura.

Cabe agregar que en esta etiqueta se puede modificar la visualización de la misma con respecto al color de los puntos, la orientación, alineación, grosor del borde, etc.

Figura 62 Monitoreo Tiempo real - Configuración yAxis para la gráfica de temperatura

```

yAxis: {
  title: {
    text: 'Valores en Centígrados'
  },
  plotLines: [{
    value: 0,
    width: 1,
    color: '#FF0000'
  }]
},
tooltip: {
  valueSuffix: ' C '
},
legend: {
  layout: 'vertical',
  align: 'right',
  verticalAlign: 'middle',
  borderWidth: 0
},
series: [{
  name: 'Temperatura',
  data: [
    <?php
      $query = " select temperatura from sensor_hyt order by id desc limit 20 ";
      $resultados = mysqli_query($connection, $query);
      while($rows = mysqli_fetch_array($resultados)){
    ?>
    <?php echo $rows["temperatura"]?>,
    <?php
      }
    ?>
  ]
}

```

Fuente: El Autor

Esta misma consulta se realiza para la gráfica de temperatura, ya que no cambian prácticamente en nada, sino solamente en el valor que se quiere mostrar en ella. A diferencia de la gráfica temperatura y humedad, en esta se visualizan las dos variables en paralelo y para

solamente toca agregar una nueva fila en yAxis el cual corresponda a la misma consulta SQL tal como se visualiza en la siguiente figura.

Figura 63 Monitoreo Tiempo real - Series de la gráfica en paralelo temperatura y humedad

```
series: [{
  name: 'Humedad',
  data: [
    <?php
      $query = " select humedad from sensor_hyt order by id desc limit 20 ";
      $resultados = mysqli_query($connection, $query);
      while($rows = mysqli_fetch_array($resultados)){
        ?>
        <?php echo $rows["humedad"]?>,
        <?php
      }
    ?>
  ]
}, {
  name: 'Temperatura',
  data: [
    <?php
      $query = " select temperatura from sensor_hyt order by id desc limit 20 ";
      $resultados = mysqli_query($connection, $query);
      while($rows = mysqli_fetch_array($resultados)){
        ?>
        <?php echo $rows["temperatura"]?>,
        <?php
      }
    ?>
  ]
}]
```

Fuente: El Autor

Para finalizar con el modulo se realiza la maquetación en el archivo monitoreo.php el cual lo que se realiza es la creación de una etiquetas HTML que se llama iframe para invocar las gráficas y se visualicen de forma correcta en el sistema.

Figura 64 Monitoreo Tiempo real - Incluir archivos en Iframe para su visualización

```
<section class="content">

<iframe src="../../../secadero/vistas/grafico/grafico_temperatura.php" id="myFrame" frameborder="0"></iframe>
<br />
<iframe src="../../../secadero/vistas/grafico/grafico_humedad.php" id="myFrame" frameborder="0"></iframe>
<br />
<iframe src="../../../secadero/vistas/grafico/grafico.php" id="myFrame" frameborder="0"></iframe>

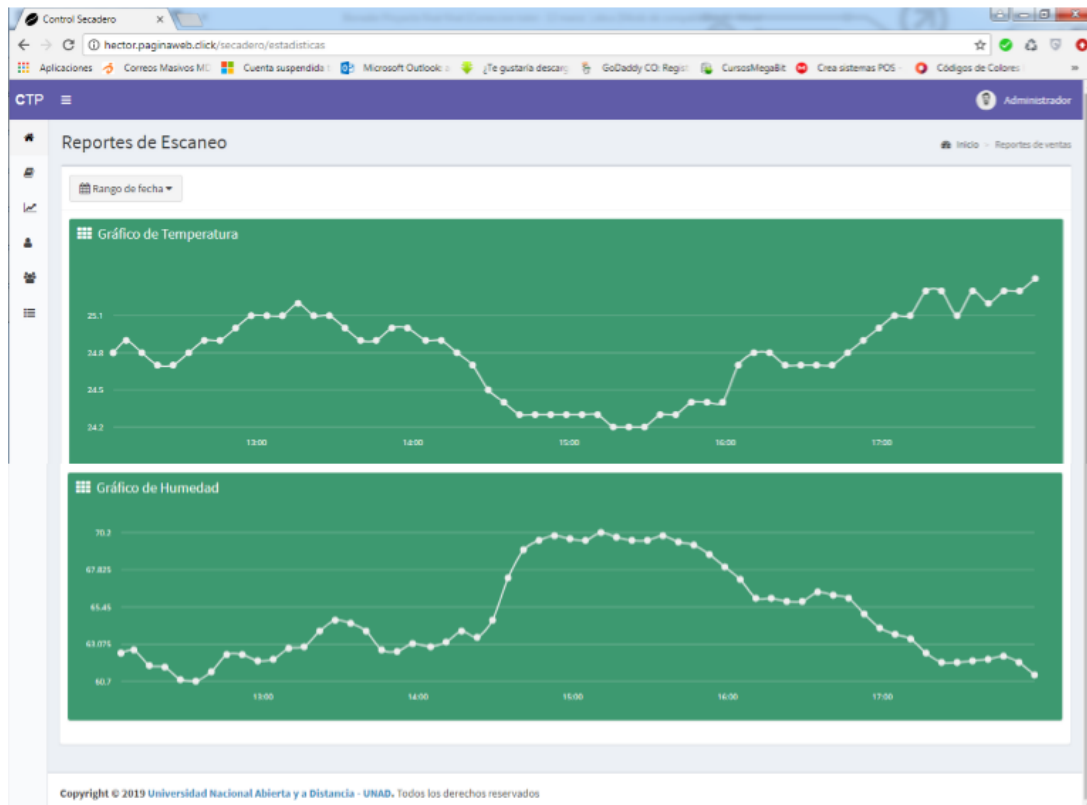
</section>
```

Fuente: El Autor

Informes – Estadísticas

En este módulo se encuentran unas graficas el cual almacena todos los registros que se encuentran en la base de datos del sistema pero solamente muestran los últimos 60 registros, el cual esto equivalen a 6 horas de monitoreo de secado. Estas gráficas fueron desarrolladas no con Highcharts sino con Morris.js y simplemente en el archivo de maquetación se le agregaron unas clases para mejorar si visualización.

Figura 65 Módulo de Estadísticas



Fuente: El Autor

Esta grafica se trabaja muy similar a la de Highcharts puesto que también toca asociarla a los archivos de la documentación principal de ellos y cuenta con una gran cantidad de gráficos, como lo son: barras, donas (donut), líneas, área, etc. Ya sabiendo esto solamente se enfocara en la parte de la programación del módulo.

Inicialmente se tiene un archivo que se llama estadísticas.php en este documento se realiza la maquetación con las etiquetas HTML que todo programador conoce y allí se crea una clase box para posicionar lo solicitado. En el box-header se agrega un botón con el identificador daterange-btn el cual es un complemento que trae la plantilla AdminLTE y este se asocia en el archivo plantilla.php para poder obtener los beneficios que este ofrece.

Figura 66 Estadísticas - Botón daterange

```
<button type="button" class="btn btn-default" id="daterange-btn">  
  <span>  
    <i class="fa fa-calendar"></i> Rango de fecha  
  </span>  
  <i class="fa fa-caret-down"></i>  
</button>
```

Fuente: El Autor

Para programar la funcionalidad del botón se tiene que realizar con JavaScript para ello se creó un archivo llamado estadísticas.js en donde va ser agregado el código que proporciona el frameworks Date Range Picker ya que es un complemento como Morris.js y Highcharts.

Figura 67 Estadísticas - Botón Daterange Picker



Fuente: El Autor

Entonces ahora se enfoca en el archivo de programación js que tiene las siguientes líneas de código.

Figura 68 Estadísticas - Configuración botón daterange

```

/*=====
RANGO DE FECHAS
=====*/
$('#daterange-btn').daterangepicker(
{
  ranges : {
    'Hoy' : [moment(), moment()],
    'Ayer' : [moment().subtract(1, 'days'), moment().subtract(1, 'days')],
    'Últimos 7 días' : [moment().subtract(6, 'days'), moment()],
    'Últimos 30 días' : [moment().subtract(29, 'days'), moment()],
    'Este mes' : [moment().startOf('month'), moment().endOf('month')],
    'Último mes' : [moment().subtract(1, 'month').startOf('month'), moment().subtract(1, 'month').endOf('month')]
  },
  startDate: moment(),
  endDate : moment()
},

function (start, end) {
  $('#daterange-btn span').html(start.format('MMM D, YYYY') + ' - ' + end.format('MMM D, YYYY'));

  var fechaInicial = start.format('YYYY-MM-DD');
  var fechaFinal = end.format('YYYY-MM-DD');
  var capturarRango = $("#daterange-btn span").html();
  localStorage.setItem("capturarRango", capturarRango);

  window.location = "index.php?ruta=estadisticas&fechaInicial="+fechaInicial+"&fechaFinal="+fechaFinal;
}
)

```

Fuente: El Autor

Lo único que se agrega en esta ilustración es la creación de las variables fechaInicial, fechaFinal (para cambiar los rangos de fecha) y capturar rango (captura el rango que se genera por el botón el cual está asociado por el id daterange-btn span) la intención con esto es que cuando se seleccione un rango de fecha con el botón, entonces se envíe a través de una variable GET ese rango de fecha, el cual va hacer tanto la fecha inicial como la fecha final. Por eso se almaceno la variable capturar rango en el localStorage.

Debido a esto se crear al inicio del documento se un filtro para saber si viene esta variable capturarRango, en donde existe una condición la cual indica que si la variable capturarRango es diferente de nulo, entonces se va colocar en el span del botón daterange lo que este guardado en

su momento de esa variable. Y si no se cumple esa condición entonces en ese botón daterange se coloca el icono y el texto inicial que está en la maqueta del archivo estadísticas.php

Figura 69 Estadísticas – LocalStorage

```
/*=====
VARIABLE LOCAL STORAGE
=====*/

if(localStorage.getItem("capturarRango") != null){
    $("#daterange-btn span").html(localStorage.getItem("capturarRango"));
}else{
    $("#daterange-btn span").html('<i class="fa fa-calendar"></i> Rango de fecha')
}
}
```

Fuente: El Autor

Ya haciendo esto se puede colocar un Windows location el cual será el encargado de enviar la variable GET y se concatenan las variable fecha inicial y fecha final. Posteriormente para el botón cancelar del daterange se realiza el siguiente código. El cual tiene como objetivo principal remover la variable GET que esta almacenada en el daterange y recarga la página a estadísticas.php

Figura 70 Estadísticas - LocalStorage Opción cancelar

```
/*=====
CANCELAR RANGO DE FECHAS
=====*/

$(".daterangepicker .range_inputs .cancelBtn").on("click", function(){
    localStorage.removeItem("capturarRango");
    localStorage.clear();
    window.location = "estadisticas";
})
}
```

Fuente: El Autor

Solamente hace falta capturar la fecha “Hoy”, para esto se tiene que identificar en donde está almacenado la opción “hoy” el cual está en las clases daterangepicker → ranges → li para que se obtenga estos datos mediante un evento click. Luego se declara una variable el cual obtiene el atributo “hoy” y este se iguala en una condición con el objetivo de declarar unas variables que van a obtener la fecha por independiente y así cumplir con los requerimientos exigidos cuando el mes es menor que 10, el día menor que 10 y el mes y el día menos que 10. Por ultimo a capturar rango se le asigna la palabra “hoy” para cuando se recargue la página y se envía el Windows location.

Figura 71 Estadísticas - Capturando fecha "Hoy"

```
$(".daterangepicker .ranges li").on("click", function(){  
    var textoHoy = $(this).attr("data-range-key");  
    if(textoHoy == "Hoy"){  
        var d = new Date();  
        var dia = d.getDate();  
        var mes = d.getMonth()+1;  
        var año = d.getFullYear();  
        if(mes < 10){  
            var fechaInicial = año+"-"+mes+"-"+dia;  
            var fechaFinal = año+"-"+mes+"-"+dia;  
        }else if(dia < 10){  
            var fechaInicial = año+"-"+mes+"-"+dia;  
            var fechaFinal = año+"-"+mes+"-"+dia;  
        }else if(mes < 10 && dia < 10){  
            var fechaInicial = año+"-"+mes+"-"+dia;  
            var fechaFinal = año+"-"+mes+"-"+dia;  
        }else{  
            var fechaInicial = año+"-"+mes+"-"+dia;  
            var fechaFinal = año+"-"+mes+"-"+dia;  
        }  
        localStorage.setItem("capturarRango", "Hoy");  
        window.location = "index.php?ruta=estadisticas&fechaInicial="+fechaInicial+"&fechaFinal="+fechaFinal;  
    }  
})
```

Fuente: El Autor

En este punto ya el funcionamiento del Date range picker esta compeltado, solamente hace falta detallar la configuración del archivo grafico_reporte_temperatura.php. El cual contiene una condición inicial que consiste en saber si existe una variable GET fechaInical en el sistema entonces se va almacenar las dos variables, fechaIncial y fechaFinal, y si no se cumple esta condición entonces esas dos variables van a ser nulas. Seguidamente se crea la clase ControladorInformes y el metodo ctrRangoFechas donde s epasan las dos variables.

Figura 72 Estadísticas - Condición para fechaIncial y fechaFinal

```
<?php
error_reporting(0);
if(isset($_GET["fechaInicial"])){
    $fechaInicial = $_GET["fechaInicial"];
    $fechaFinal = $_GET["fechaFinal"];
}else{
    $fechaInicial = null;
    $fechaFinal = null;
}
$respuesta = ControladorInformes::ctrRangoFechas($fechaInicial, $fechaFinal);
```

Fuente: El Autor

Ahora en estadísticas.controlador.php se encuentra este método que consiste en recibir las dos variables anteriormente definidas y además se crea una nueva variable la cual se le asigna la tabla de sensor_hyt (es donde están almacenados los valores obtenidos con el prototipo) y también es creada la clase ModeloInformes y el metodo mdlRangoFechas, donde se pasan las 3 variables que se acaban de mencionar.

Figura 73 Estadísticas - Controlador Rango Fechas

```
/*=====
RANGO FECHAS
=====*/

static public function ctrRangoFechas($fechaInicial, $fechaFinal){

    $tabla = "sensor_hyt";
    $respuesta = ModeloInformes::mdlRangoFechas($tabla, $fechaInicial, $fechaFinal);
    return $respuesta;

}
```

Fuente: El Autor

Por ultimo en el archivo estadísticas.modelo.controlador.php se encuentra el único método generado que corresponde a la consulta de la base de datos para las gráficas de estadísticas este se separa en tres (3) momentos.

Primero consiste en que si fechaInicial viene nula se imprime todo lo que tenga que ver con las fechas del sistema, en pocas palabras se trae todos los resultados de la tabla.

Segundo puede suceder que al seleccionar la opción “hoy” entonces las fechas van hacer iguales, por eso la forma que se utilizó para capturar la fecha es preguntar que si en la tabla sensor_hyt de la columna fecha_hora que coincidencias encuentra con respecto a la fechaFinal.

Finalmente se crea una variable fechaActual para capturar el día de “hoy” y a esa fecha actual se le adiciona un día más con **add(new DateInterval("P1D"))** luego es creada la fechaActualMasUno el cual se captura de la siguiente manera **\$fechaActual->format("Y-m-d")** lo mismo se realiza con la fechaFinal2. Entonces si fechaFinalMasUno es igual a fechaActualMasUno entonces si es asi, se solicita de la base de datos un BETWEEN entre la fecha inicial que viene más la fechaFinalMasUno. Y si no se cumple entonces que envíe un BETWEEN entre fechaInicial y fechaFinal

Porque el datarange picker está tomando el rango de fechas de los últimos 7 días o los últimos 30 días hasta el día de “ayer” y no tiene en cuenta el día de “hoy” entonces por eso es necesario incrementar un día más para que tome hasta el día de “hoy” y así imprima todo los registros completos.

Figura 74 Estadísticas - Modelo Rango Fechas

```

static public function mdlRangoFechas($tabla, $fechaInicial, $fechaFinal){

    if($fechaInicial == null){

        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla ORDER BY id DESC LIMIT 120");
        $stmt -> execute();
        return $stmt -> fetchAll();

    }else if($fechaInicial == $fechaFinal){

        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE fecha_hora like '%$fechaFinal%'");
        $stmt -> bindParam(":fecha_hora", $fechaFinal, PDO::PARAM_STR);
        $stmt -> execute();
        return $stmt -> fetchAll();

    }else{

        $fechaActual = new DateTime();
        $fechaActual ->add(new DateInterval("P1D"));
        $fechaActualMasUno = $fechaActual->format("Y-m-d");

        $fechaFinal2 = new DateTime($fechaFinal);
        $fechaFinal2 ->add(new DateInterval("P1D"));
        $fechaFinalMasUno = $fechaFinal2->format("Y-m-d");

        if($fechaFinalMasUno == $fechaActualMasUno){

            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE fecha_hora BETWEEN '$fechaInicial' AND '$fechaFinalMasUno'");

        }else{

            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE fecha_hora BETWEEN '$fechaInicial' AND '$fechaFinal'");

        }

        $stmt -> execute();
        return $stmt -> fetchAll();

    }

}
  
```

Fuente: El Autor

Ya teniendo estos resultados óptimos para visualizar los datos en la gráfica entonces se necesita agrupar los datos por fechas y que se visualicen en de forma dinámica en el sistema. Por eso lo primero que se realiza es la declaración de unas variables en forma de array como lo son: arrayFechas, arrayTemperatura y nuevaTemperatura.

Seguidamente se crea un foreach para hacer un recorrido asociativo sobre la variable \$respuesta por todos los índices. Primero se captura la fecha que está en sistema, luego se introduce esa variable \$fecha en arrayfechas, cuando ya se tiene esto entonces se introduce la temperatura en un array el cual es \$arraytemperatura y finalmente se imprimen en un foreach los valores de la variable \$nuevaTemperatura por los índices [\$key] y seguidamente se crea una variable llamada \$norepetirDatos el cual se utiliza la función **array_unique** para que no se repitan las fecha.

Figura 75 Estadísticas - Array para agrupar datos

```
$arrayFechas = array();  
$arrayTemperatura = array();  
$nuevaTemperatura = array();  
  
foreach ($respuesta as $key => $value) {  
    $fecha = substr($value["fecha_hora"],0,16);  
    array_push($arrayFechas, $fecha);  
    $arrayTemperatura = array($fecha => $value["temperatura"]);  
    foreach ($arrayTemperatura as $key => $value) {  
        $nuevaTemperatura[$key] = $value;  
    }  
}  
  
$noRepetirDatos = array_unique($arrayFechas);  
?>
```

Fuente: El Autor

Cuando se ha realizado todo este procedimiento tan largo es momento de ir al código que proporciona Morris.js para mostrar los datos en la gráfica de forma dinámica. Según la documentación de Morris tiene 5 elementos que son requeridos y obligatorios para poder visualizar la gráfica, estos son:

- element: Es el ID asociativo que se coloca en el HTML, este elemento debe tener por obligación un ancho y altos definidos con su propio estilo.

Figura 76 Estadísticas – Element

```
<div class="chart" id="line-chart-informes" style="height: 250px;"></div>
```

Fuente: El Autor

- data: es donde se realiza la consulta php para visualizar los datos en la gráfica de forma dinámica. En este caso se aplicó un foreach con respecto a la variable \$noRepetirDatos.

Figura 77 Estadísticas – Data

```
data : [
<?php
if($noRepetirDatos != null){
    foreach ($noRepetirDatos as $key) {
        echo "{ fecha: '" . $key . "', humedad: '" . $nuevaTemperatura[$key] . "' },";
    }
    echo "{ fecha: '" . $key . "', humedad: '" . $nuevaTemperatura[$key] . "' }";
} else {
    echo "{ fecha: '0', humedad: '0' }";
}
?>
],
```

Fuente: El Autor

- xkey: Una cadena que contiene el nombre del atributo que contiene los valores de fecha (X).
- ykeys: Una lista de cadenas que contienen nombres de atributos que contienen valores

- labels: Una lista de cadenas que contienen etiquetas para las series de datos a trazar
(correspondientes a los valores en la ykeys opción)

En este caso se aplicó lo solicitado a los valores requeridos de Morris.js lo único diferente es que se agregó una etiqueta personalizada el cual indica que si la temperatura sobre pasa los 45 grados, entonces informar en un label que ha activado el ventilados para bajar la presión”

Figura 78 Estadísticas - Etiquetas para la gráfica Morris js

```

],
xkey      : 'fecha',
ykeys     : ['temperatura'],
labels    : ['temperatura'],
lineColors : ['#efefef'],
hoverCallback : function (index, options, content, row) {
  if (row.temperatura > 45) {
    return "Se activo el Ventilador para bajar la presión <br> Fecha: " + row.fecha + " <br>
      temperatura (" + row.temperatura + ")";
  }else if(row.temperatura < 45){
    return "Fecha: " + row.fecha + " <br> temperatura (" + row.temperatura + ")";
  }
},
lineWidth : 2,
hideHover : 'auto',
gridTextColor : 'fff',
gridStrokeWidth : 0.4,
pointSize : 4,
pointFillColors : ['#efefef'], //aquí es donde se cambia el color de los puntos
pointStrokeColors: ['#efefef'], //aquí se cambia el color de la circunferencia del punto
gridLineColor : '#efefef',
gridTextFamily : 'Open Sans',
gridTextSize : 10,
ymax : 'auto',
ymin : 'auto'

```

Fuente: El Autor

Para concluir este mismo procedimiento se realizó a la gráfica de humedad a diferencia de los parámetros que se utilizan solamente en el archivo grafico_reporte_humedad.php. En cuanto al controlador y el modelo si se aplica el mismo, por ende no hay necesidad de crear uno nuevo.

Informes - Reportes

En este módulo se encontraran los registros de la base de datos agrupados por fecha, donde se relaciona la cantidad de registros de esa fecha, se podrá visualizar esta cantidad por individual si así lo desea y también podrá consultar el promedio de temperatura y humedad que tiene esa fecha, además de revisar los datos máximos y mínimos de temperatura y humedad de esa fecha.

Figura 79 Informes – Reportes

Fecha	Registros	Acciones
2019-03-03	159	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-04	137	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-05	167	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-06	5	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-07	42	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-08	91	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-09	240	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-10	229	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos
2019-03-11	240	Registros Promedio Temperatura y Humedad Datos Maximos y Minimos

Fuente: El Autor

Ahora se pasa al archivo principal el cual es reportes.php en este archivo se realiza la maquetación con las etiquetas HTML conocidas por todo programador. Para ello primero que todo se agrega una tabla con los elementos solicitados que son: Fecha, Registros y Acciones.

Entonces primero se tiene que realizar una conexión PDO con la base de datos del servidor, el cual se puede visualizar en la siguiente figura los campos que se solicitan para la conexión.

Figura 80 Reportes – Conexión PDO

```
<?
$server="localhost";
$database="hectorelfather_control_secadero";
$dbpass="H3f@dumo1509-";
$dbuser="hectorelfather_hefadumo15";
date_default_timezone_set('America/Bogota');
?>
```

Fuente: El Autor

Luego en el archivo principal de reportes se incluye el archivo conexion_pdo.php, seguidamente de realizar la creación de una variable el cual tiene como objetivo conectar a la base de datos y sencillamente se crea un foreach en donde se aplica una consulta a la base de datos que consiste en lo siguiente.

Primero se obtiene la fecha de la columna fecha_hora en la base de datos, luego realiza un conteo de todos los valores que ha encontrado de la tabla sensor_hyt, estos valores son agrupados por fecha_hora y ordenamos de forma Ascendente por la columna fecha_hora. Si esta consulta es correcta, entonces es almacenada en la variable \$M3.

Figura 81 Informes - Reportes - Consulta para mostrar registros

```
foreach($pdo->query("SELECT date(fecha_hora) as fecha, count(*) as conteo FROM sensor_hyt group by date(fecha_hora)
order by date(fecha_hora) ASC") as $M3) {
```

Fuente: El Autor

Por último se imprime con un echo las etiquetas HTML y en la columna de fecha se adiciona un href con la siguiente ruta “reporte.php?fecha=” debido que la los registros se obtienen por el metodo GET. En cuanto a los botones se tiene que agregar una ruta como la siguiente “index.php?ruta=reporteregistros&fecha=”. Cabe agregar que para mostrar los resultados en la tabla se tiene que concatenar de la siguiente manera “\$M3['nombre del valor devuelto en la consulta’]”.

Figura 82 Informes - Reportes - maquetación tabla de reportes

```

echo '<tr>

<td><center><a href="reporte.php?fecha='.$M3['fecha'].'"></a>'.$M3['fecha'].'
</center></td>
<td><center>'.$M3['conteo'].'</center></td>

<td>
<center>
<div class="btn-group">

<a href="index.php?ruta=reporteregistros&fecha='.$M3['fecha'].'"><button class="btn btn-info"><i
class="fa fa-calendar"></i> Registros </button></a>

<a href="index.php?ruta=temperaturaregistros&fecha='.$M3['fecha'].'"> <button class="btn
btn-success"><i class="fa fa-bar-chart-o"></i> Promedio Temperatura y Humedad</button> </a>

<a href="index.php?ruta=maximominimoregistros&fecha='.$M3['fecha'].'"> <button class="btn
btn-warning"><i class="fa fa-retweet"></i> Datos Maximos y Minimos</button> </a>

</div>
</center>
</td>

</tr>';
  
```

Fuente: El Autor

Ahora para la acción mostrar registros de ese grupo de fecha, se procede en ese documento inicialmente declarar una variable que se llama \$conteo igualada a cero y luego se incluye el archivo con la información para realizar la conexión. Seguidamente se declara la variable que realiza la conexión a la base de datos y la hace efectiva.

De esta manera ya teniendo la conexión establecida se procede a realizar una condición que consiste en que si viene una variable GET “fecha” entonces \$fecha es igual a la fecha que

viene por el método GET y se almacena. Si esto no es correcto, entonces envía un mensaje diciendo que “No se envió la fecha”

Figura 83 Informes - Reportes - Método GET fecha

```
$conteo=0;

include("conexion_pdo.php");

$pdo = new PDO("mysql:host=$server;dbname=$database", $dbuser, $dbpass);

if (isset ($_GET['fecha'])) {$fecha= addslashes(strip_tags(trim($_GET['fecha'])));} else { echo 'No Enviaste fecha'; exit; }
```

Fuente: El Autor

Teniendo en cuenta esta información se realizan las etiquetas HTML las cuales diseñan la tabla en que se va a visualizar los datos correspondientes solicitados. Para ello se realiza una consulta SQL que consiste en seleccionar los campos temperatura y humedad el cual de ello se substrahe la **fecha** de la columna fecha_hora, se substrahe tambien la **hora** de la columna fecha_hora de la tabla sensor_hyt CUANDO fecha_hora sea igual a fecha y este ordenado por id de forma Descendente. Si todo resulta correcto entonces se almacena esta consulta en la variable \$M3. Para finalizar esta parte solamente hace falta concatenar la variable e imprimir el resultados que se quieren mostrar, el cual es de la siguiente manera \$M3[“nombre de la variable a mostrar.”].

Figura 84 Informes - Reportes - Consulta SQL para ingresar los registros

```
foreach($pdo->query("SELECT temperatura, humedad, date(fecha_hora) as fecha, time(fecha_hora) as hora FROM sensor_hyt WHERE date(fecha_hora)='$fecha' ORDER BY id DESC") as $M3) {

    $conteo++;

    echo '<tr>

        <td><center>'. $conteo. '</center></td>

        <td><center>'. $M3['temperatura']. '</center></td>

        <td><center>'. $M3['humedad']. '</center></td>

        <td><center>'. $M3['fecha']. '</center></td>

        <td><center>'. $M3['hora']. '</center></td>

    </tr>';

}
```

Fuente: El Autor

De los planteamientos anteriores cabe resaltar que esto se hace lo mismo tanto para el archivo de promediotemperatura.php y el archivo maximominimoregistros.php a excepto de la consultas que son totalmente diferentes ya que los resultados a mostrar tambien son diferentes. A continuación se visualizan estas consultas.

En esta consulta la única diferencia es la sentencia SQL AVG el cual tiene como función obtener el promedio de un grupo de datos.

Figura 85 Informes - Reportes - Consulta SQL para el promedio

```
foreach($pdo->query("SELECT AVG(temperatura) AS promedio_temperatura, AVG(humedad) AS promedio_humedad, date(fecha_hora) AS fecha FROM sensor_hyt WHERE date(fecha_hora)='$fecha' ORDER BY id DESC") as $M3) {

    $conteo++;

    echo '<tr>
        <td><center>'.$conteo.'</center></td>
        <td><center>'.$M3['promedio_temperatura'].'</center></td>
        <td><center>'.$M3['promedio_humedad'].'</center></td>
        <td><center>'.$M3['fecha'].'</center></td>
    </tr>';
}
```

Fuente: El Autor

Como puede observarse en esta consulta también existe un mínimo de diferencia el cual es la sentencia SQL MAX y MIN, el cual la primera se encarga de obtener el valor máximo de un grupo de datos y la segunda se obtiene el valor mínimo de un grupo de datos.

Figura 86 Informes - Reportes - Consulta SQL para los datos máximos y mínimos

```
foreach($pdo->query("SELECT max(temperatura) as maxima_tem, min(temperatura) as minima_tem, max(humedad) as maxima_hum, min(humedad) as minima_hum FROM sensor_hyt WHERE date(fecha_hora)='$fecha' ORDER BY id DESC") as $M3) {

    $conteo++;

    echo '<tr>
        <td><center>'.$conteo.'</center></td>
        <td><center>'.$M3['maxima_tem'].'</center></td>
        <td><center>'.$M3['minima_tem'].'</center></td>
        <td><center>'.$M3['maxima_hum'].'</center></td>
        <td><center>'.$M3['minima_hum'].'</center></td>
    </tr>';
}
```

Fuente: El Autor

Los resultados que se obtienen en cada una de las vistas son las siguientes.

Informes – Reportes – Registros

Figura 87 Informes - Reportes – Registros

Control Secadero

hector.paginaweb.click/secadero/index.php?ruta=reporteregistros&fecha=2019-03-03

CTP

Administrador

Registros Secadero del **2019-03-03**

Inicio > Informes > Reportes > Reportes Registros

Regresar

Mostrar 10 registros

Buscar:

#	Temperatura	Humedad	Fecha	Hora
1	22.00	70.80	2019-03-03	23:41:30
2	22.00	70.60	2019-03-03	23:35:31
3	22.00	70.50	2019-03-03	23:29:27
4	22.10	70.50	2019-03-03	23:23:26
5	22.10	70.20	2019-03-03	23:17:26
6	22.20	70.30	2019-03-03	23:11:30
7	22.20	70.30	2019-03-03	23:05:23
8	22.20	70.20	2019-03-03	22:59:19
9	22.20	70.20	2019-03-03	22:53:16
10	22.30	70.30	2019-03-03	22:47:20

Mostrando registros del 1 al 10 de un total de 159

Anterior 1 2 3 4 5 ... 16 Siguiete

Fuente: El Autor

Informes – Reportes – Promedio Temperatura y Humedad

Figura 88 Informes - Reportes - Promedio temperatura y humedad

Control Secadero

hector.paginaweb.click/secadero/index.php?ruta=temperaturaregistros&fecha=2019-03-03

CTP

Administrador

Promedio de Temperatura y Humedad **2019-03-03**

Inicio > Informes > Reportes > Promedio de Temperatura y Humedad

Regresar

Mostrar 10 registros

Buscar:

#	Promedio Temperatura	Promedio Humedad	Fecha
1	24.440252	67.325157	2019-03-03

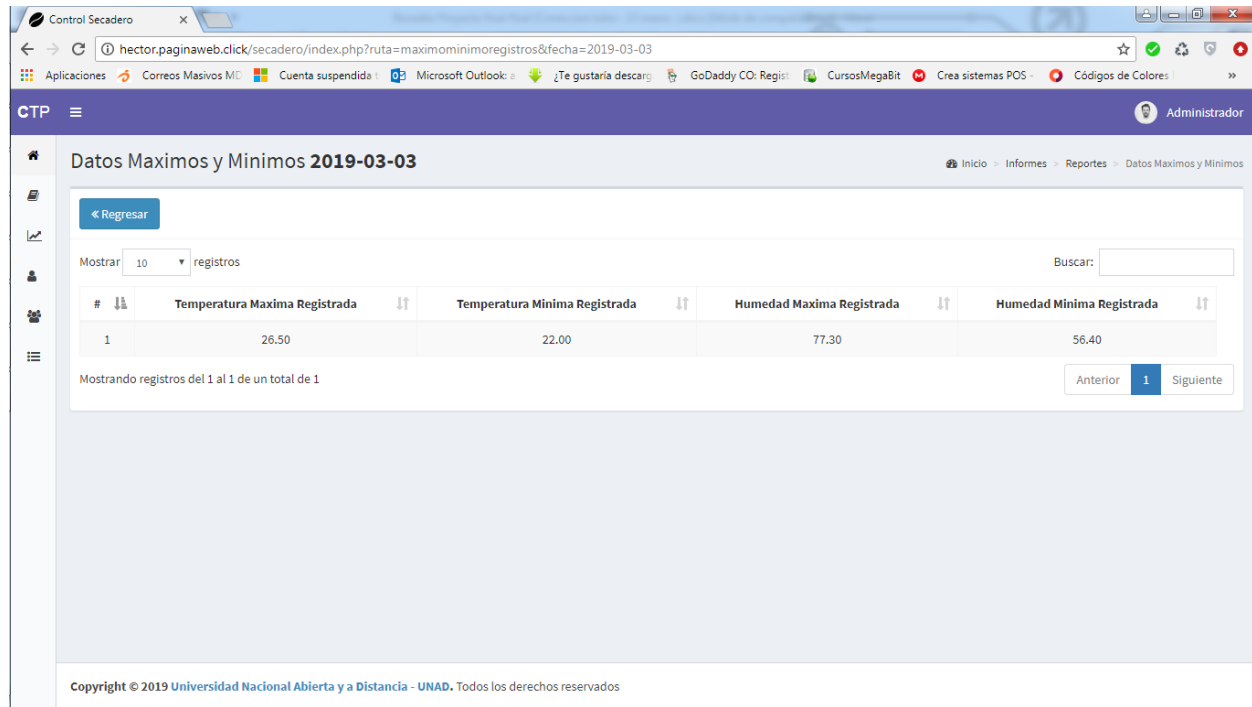
Mostrando registros del 1 al 1 de un total de 1

Anterior 1 Siguiete

Fuente: El Autor

Informes – Reportes – Datos Máximos y Mínimos

Figura 89 Informes - Reportes - Datos Máximos y Mínimos



Control Secadero

hector.paginaweb.click/secadero/index.php?ruta=maximominimoregistros&fecha=2019-03-03

CTP

Administrador

Datos Maximos y Minimos 2019-03-03

Inicio > Informes > Reportes > Datos Maximos y Minimos

<< Regresar

Mostrar 10 registros

Buscar:

#	Temperatura Maxima Registrada	Temperatura Minima Registrada	Humedad Maxima Registrada	Humedad Minima Registrada
1	26.50	22.00	77.30	56.40

Mostrando registros del 1 al 1 de un total de 1

Anterior 1 Siguiente

Copyright © 2019 Universidad Nacional Abierta y a Distancia - UNAD. Todos los derechos reservados

Fuente: El Autor

Para finalizar el sistema solamente hacen faltan 2 módulos, el de Inicio y el de noticia. Se dejaron estos módulos de último ya que son informativos y no requieren de gran complejidad su desarrollo.

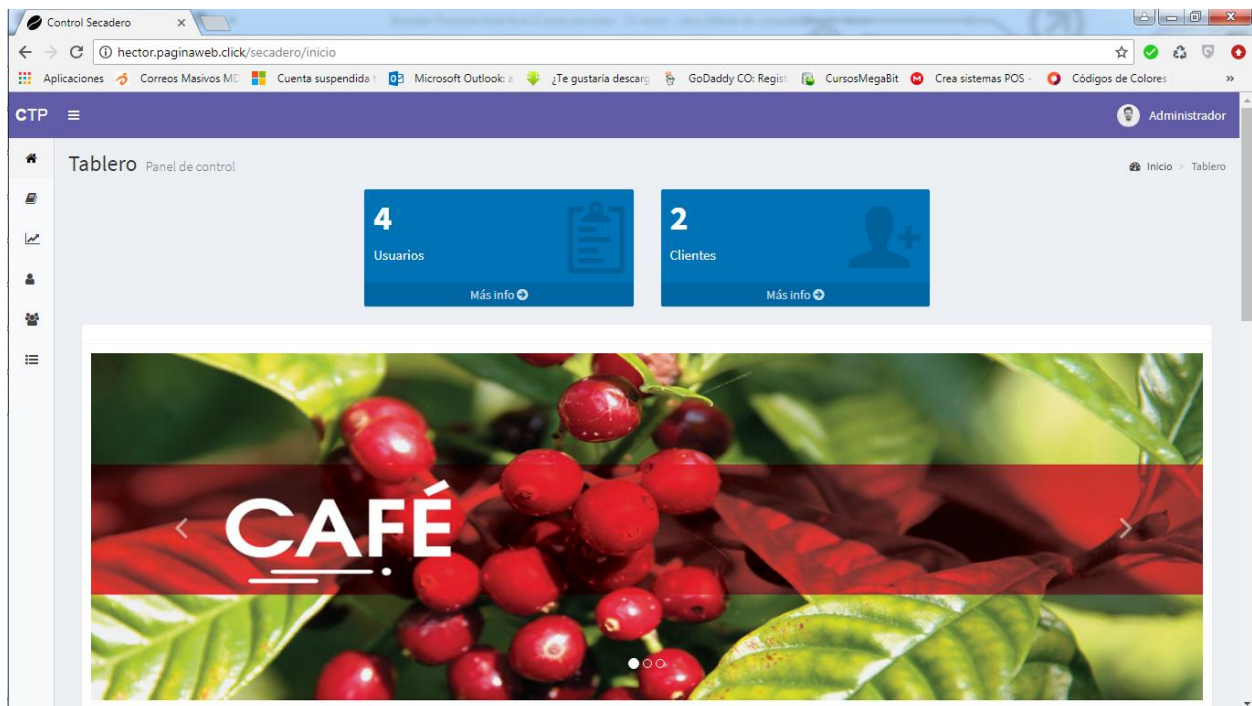
Módulo Inicio - Noticias

En este módulo se busca tener una vista amigable con respecto al cliente, en donde el administrador puede visualizar todo el contenido del sistema y además tiene en su página de inicio 2 cajas principales el cual le informan cuantos usuarios están registrados y cuantos clientes están registrados en la base de datos. Esta información solamente el podrá visualizarla el administrador aquí se empieza a ver la diferencia de los roles y el control que se tiene en el sistema es un tema que más adelante se dará a conocer. Por otra parte en el módulo de inicio se

presenta un slider con imágenes alusivas al cultivo del café, métodos de cosecha, pos-producción, etc.

A demás se incluirá más abajo unas noticias informativas con respecto al mismo tema, pero en este caso no solamente se visualizara un fragmento de la noticia y luego un botón “leer más” lo direccionara al módulo de noticias.

Figura 90 Módulo de Inicio



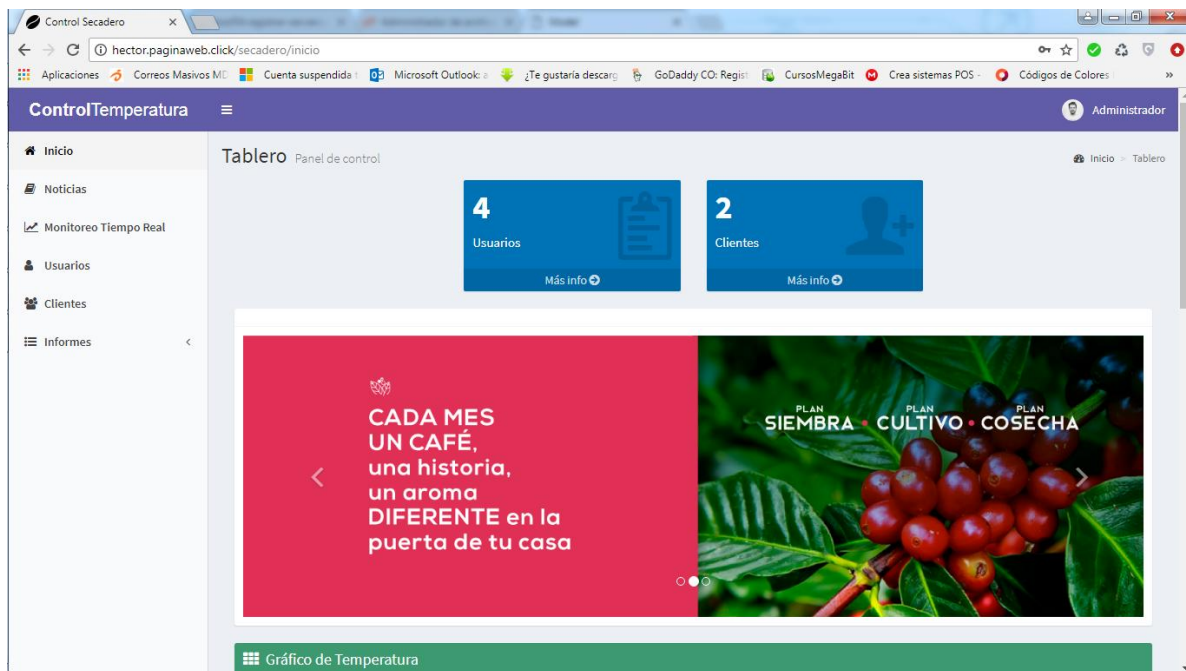
Fuente: El Autor

Para finalizar como se ha mencionado antes, en el módulo de noticias solamente es informativo y de interés con respecto al producto que se está tratando en el sistema, el usuario puede encontrar gran variedad de información que le puede servir de provecho para su cultivo y enriquecer su conocimiento, ya que como todo mundo sabe muchas los caficultores carecen de información esencial y actual sobre como tener un cultivo de calidad.

Para finalizar con lo propuesto se resalta que en el sistema cuenta con una serie de roles que serán ejecutados de forma óptima para llevar un orden en la plataforma. Donde en el capítulo 4 y 5 se hace una detallada información sobre estos roles y las funciones que cada uno puede desempeñar en el sistema. A continuación se presentaran las diferentes vistas dependiendo el rol con el que se ingrese al sistema.

Rol de Administrador: Tiene acceso a todo el contenido del sistema

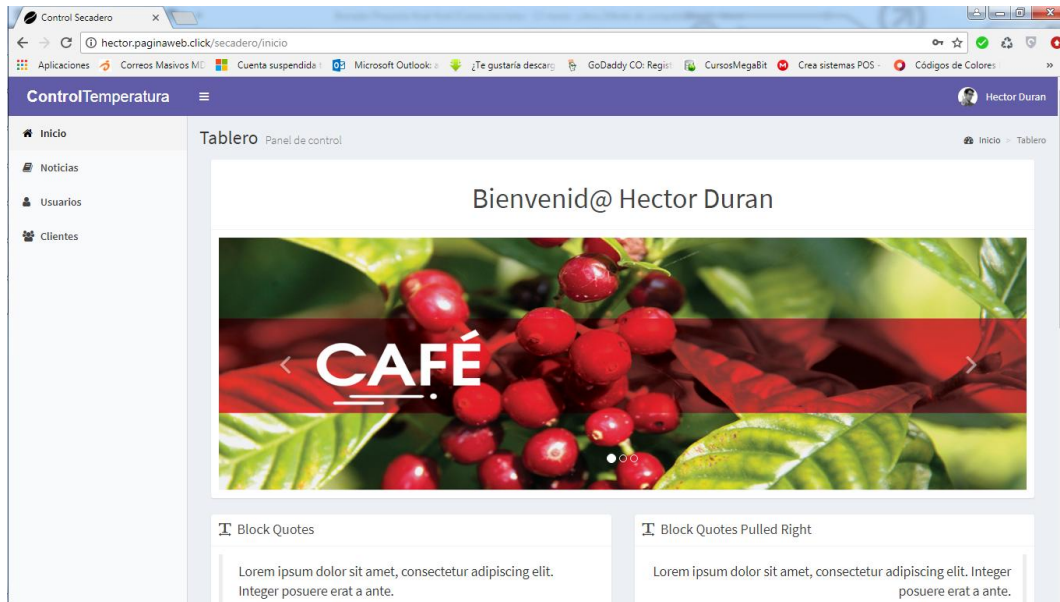
Figura 91 Inicio - Vista Rol de Administrador



Fuente: El Autor

Rol de Especial: Tiene acceso a los módulos de Inicio, Noticias, Usuarios y Clientes.

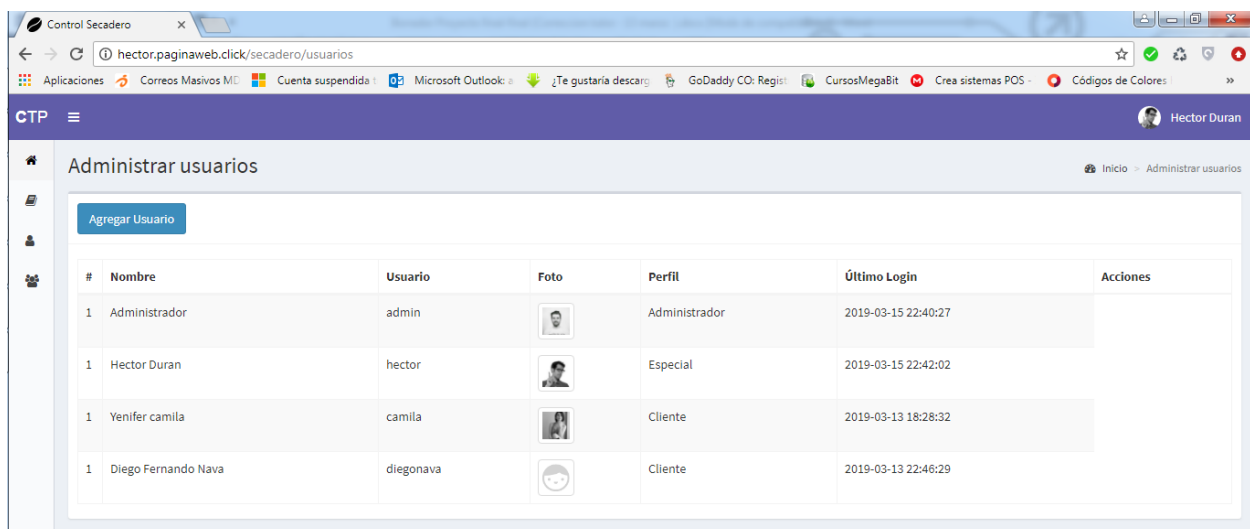
Figura 92 Inicio - Vista Rol de Especial



Fuente: El Autor

En el módulo de usuarios solamente puede agregar un usuario y visualizar los usuarios que están registrados en el sistema No podrá eliminar o editar datos.

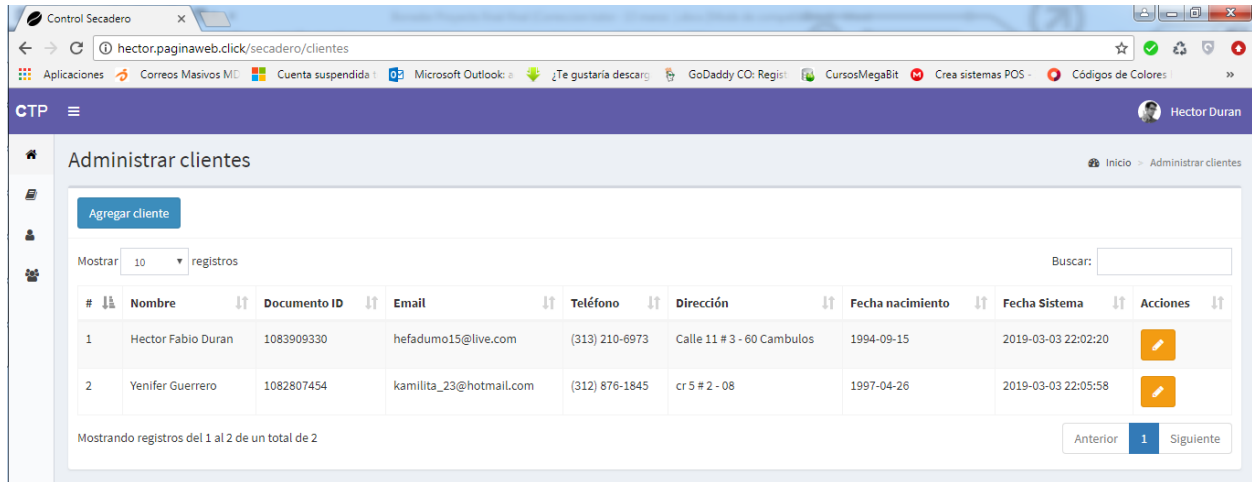
Figura 93 Inicio - Inicio - Vista Rol de Especial - Administrador de usuarios



Fuente: El Autor

En el módulo de Clientes el rol de especial puede crear un nuevo cliente, visualizar los clientes y puede editar la información de cada uno de ellos.

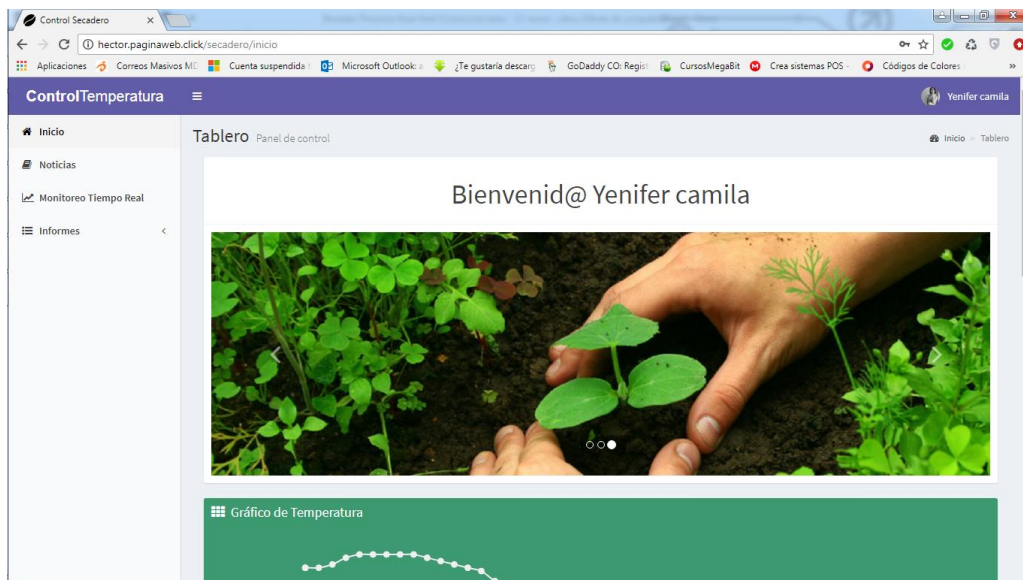
Figura 94 Inicio - Vista Rol de Especial - Administrador de Clientes



Fuente: El Autor

Rol de Cliente: Tiene acceso a los módulos de Inicio, Noticias, Monitoreo tiempo real e Informes.

Figura 95 Inicio - Vista Rol de Cliente

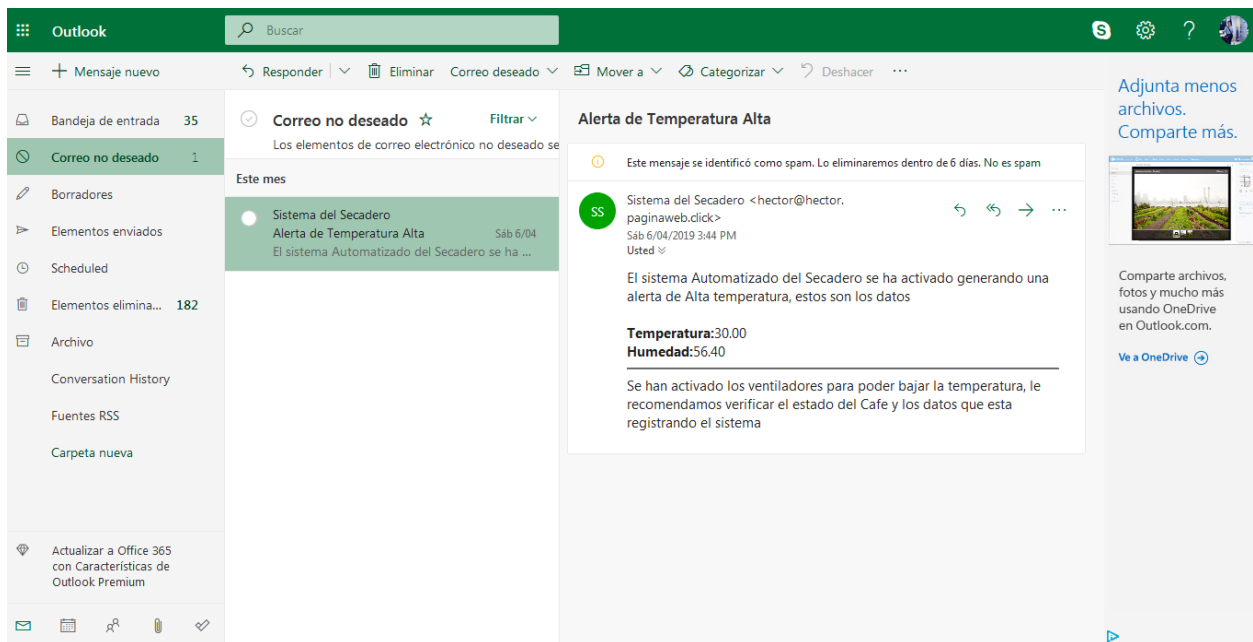


Fuente: El Autor

Por ultimo cuando el sistema detecta que se ha aumentado los valores de la temperatura y humedad relativa del ambiente, inmediatamente envía un mensaje al correo electrónico del caficultor informando las acciones realizadas por el sistema para lograr disminuir estos valores y así obtener una temperatura y humedad adecuada.

En la figura 96 se puede visualizar una prueba realizada el cual corresponde cuando es sobre pasado los 30 grados de Temperatura.

Figura 96 Alerta Generada por el sistema



Fuente: El Autor

Los archivos con el código fuente de programación se pueden visualizar en el siguiente enlace para su verificación con respecto al sistema creado.

Enlace: <https://drive.google.com/open?id=1CVQrvX1kzyl yap2BIDXMgrCuY3oRQVUx>

Indicaciones para el usuario

Esta plataforma no cuenta con un sistema de registro ya que no está abierta a cualquier persona, está enfocada en una población específica que son los caficultores, por ende tendrá que comunicarse con el administrador del sistema para que solicite la creación de un usuario y así poder ingresar al sistema.

A continuación se describe información esencial para el correcto funcionamiento del sistema. Primero en el Modulo de inicio, podrá visualizar contenido de interés con respecto a nuevas técnicas, productos, artículos sobre el proceso del café, precio del café, entre otras. Además puede visualizar en primera instancia los resultados de los últimos 60 registros que han obtenido en el sistema de ControlTemperatura el cual equivalen a 6 horas de monitoreo.

En el módulo de noticias puede encontrar información de interés con más contenido el cual puede ser de su interés.

En el módulo de Monitoreo tiempo real solamente podrá visualizar los datos en tiempo real y si pasa el cursor por encima de los puntos de la gráfica, podrá ver en detalle la temperatura o humedad que se está registrando en ese momento.

En el módulo informes encontrara dos opciones más, la primera Estadísticas. Allí podrá ingresar y visualizar de forma dinámica todos los datos obtenidos que se han registrado en el sistema y además puede consultar los resultados que desee en cualquier fecha, eso sí tiene registros. Las opciones son Hoy, ayer, últimos 7 días, últimos 30 días, este mes, último mes y rango personalizado.

En el segundo ítems Reportes, encontrara los registros agrupados por fecha y tiene la posibilidad de visualizar todos los registros de esa fecha, la cantidad de registros que se

obtuvieron, también si desea visualizar la cantidad total de registros por individual lo puede hacer. A demás puede visualizar el promedio de la temperatura y la humedad, el punto más alto de la temperatura y humedad y el punto más bajo de la temperatura y humedad.

Resultados

Se plantean 3 tipos de resultados cuando se finaliza el proyecto, los cuales son:

- La investigación: Se obtuvo como resultado un análisis de información primaria sobre el conocimiento que tiene el caficultor en relación a las diferentes nuevas tecnologías y procesos de comercialización sobre el producto del café

Se puede visualizar las encuestas en el siguiente enlace:

<https://drive.google.com/open?id=1pXicXtBNUx0YZKPCsUdoN-rNVGNkqOrZ>

- El diseño del prototipo: Inicialmente fueron planteadas varias alternativas para el prototipo el cual se eligió la más adecuada para el proyecto teniendo en cuenta recursos económicos y elaboración de la maqueta.

Se puede visualizar el desarrollo del prototipo en el siguiente enlace:

<https://drive.google.com/open?id=1XKE3jD3q9J-D7JwrDpczvSnFjThyySKe>

Se puede visualizar el prototipo Finalizado en el siguiente enlace:

<https://drive.google.com/open?id=1ZKFAWRRvaa35L1AmlzE4heWxK5SN4MrY>

- El diseño de la página web: Se desarrolló una plataforma adaptable a todos los dispositivos y cumpliendo los diferentes requerimientos, controles y módulos que exigía el proyecto.

Se puede visualizar el desarrollo de la página web en el siguiente enlace:

https://drive.google.com/open?id=1Mu0iN3k5YRZR1w_uYXlrbpRT5BHCbA3

De igual manera se comparten datos de acceso para ingresar y visualizar el contenido de la página web. (No se comparte los datos del administrador del sistema)

Enlace de la página web: <http://hector.paginaweb.click/secadero/>

- ***Rol Especial:***

Usuario: hector

Contraseña: hector123

- ***Rol Cliente:***

Usuario: diegonava

Contraseña: diegonava123

Recomendaciones

Como todo proyecto siempre hay unas recomendaciones que hacer cuando se finaliza un estudio, sin embargo el proyecto puede tener un gran futuro si se plantea a grandes rasgos de proyección.

A continuación se presentan algunas recomendaciones o mejoras para el sistema

- Implementar un cámara dentro del sistema de ControlTemperatura, con el objetivo de que el usuario pueda verificar que no hay ningún problema con el café.
- Se plantearía una propuesta para cuando no haya internet, debido que esta es la gran falencia del sistema.
- Mejorar el código de programación para que la plataforma sea no para una placa de Arduino sino para cientos de placas.
- Se puede colocar un foro Web en el sistema con el objetivo de mantener una interacción de los usuarios y se puede resolver preguntas por parte del administrador o administradores del sistema.
- Implementar en el sistema un botón el cual al dar clic se inicie con el monitoreo del producto y de la misma manera se pueda detener.

Conclusiones

Con este proyecto se ha comprendido el procedimiento y complejidad que se lleva a cabo un planteamiento desde el inicio en que se realiza la investigación, puesto que en este punto es donde el estudiante determina cuáles son los objetivos primordiales del proyecto y hacia donde se está enfocando, esto para garantizar el óptimo proceso de su elaboración, además que sea un proyecto viable e innovador en todo sentido. Por otra parte es fundamental porque enriquece el conocimiento del estudiante y el director del proyecto, el cual los lleva a indagar en profundidad del tema escogido y buscar opciones para resolver los problemas que se vayan presentando en el camino del desarrollo de este prototipo.

Al realizar la implementación de este proyecto se logró identificar que en los secaderos solares se obtienen cifras de acaloramiento superiores a 50 °C afectando directamente la salud de los caficultores, ya que en la mayoría de los casos no se implementan medidas de protección ante los altos niveles de temperatura que se obtienen en el secadero. En cuanto al software creado para el prototipo se comprende que indispensablemente mejora la calidad de vida del agricultor, porque ofrece principalmente un control y monitoreo de la temperatura y humedad del ambiente en el cual se pueden visualizar los datos en tiempo real y desde cualquier lugar que se desee (claro está teniendo en cuenta que se requiere conexión a internet) además se ofrecen reportes y estadísticas con respecto a los datos obtenidos durante el o los procesos realizados con el sistema. También se genera una alerta cuando se excede la temperatura óptima permitida en el secadero y se brinda una recomendación para que el caficultor verifique el estado de su producto.

Para terminar el software cuenta con una sección de noticias que solamente contiene información del producto del café, en donde el agricultor puede comprender nuevos métodos o técnicas para el aprovechamiento de su producto.

En definitiva se puede decir que con la implementación de este prototipo el caficultor solamente tendrá que determinar manualmente el punto exacto de secado para su café, ya que no está programado para esta tarea, porque para llegar a este punto se requieren muchos factores que son importantes para mejorar el factor de producción del café y es muy complicado determinar su valor exacto.

Referencias Bibliográficas

- ACIS. (Enero de 2016). *La evolución del Internet de las Cosas*. Recuperado el 10 de Septiembre de 2018, de Asociación Colombiana de Ingenieros de Sistemas:
<http://acis.org.co/portal/content/la-evoluci%C3%B3n-del-internet-de-las-cosas-0>
- Anacafe. (s.f.). *Asociación Nacional del Café*. Obtenido de El proceso de secado de café:
https://www.anacafe.org/glifos/index.php?title=Secadoras_Proceso
- Aprendiendo Arduino. (23 de Enero de 2013). *Programación Arduino*. Recuperado el 10 de Septiembre de 2018, de Aprendiendo Arduino:
<https://aprendiendoarduino.wordpress.com/2017/01/23/programacion-arduino-5/>
- Aprendiendo Arduino. (4 de Julio de 2016). *Ethernet Shield*. Recuperado el 12 de Septiembre de 2018, de Aprendiendo Arduino:
<https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>
- Arduino. (s.f.). *¿Que es Arduino?* Recuperado el 11 de Septiembre de 2018, de Arduino:
<http://arduino.cl/que-es-arduino/>
- Aulaformativa. (30 de Junio de 2017). *Definición, usos y ventajas del lenguaje CSS3*. Recuperado el 16 de octubre de 2018, de Aula Formativa:
<https://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>
- aulaformativa. (2 de Marzo de 2019). *Definición, usos y ventajas del lenguaje HTML5*. Obtenido de aulaformativa: <https://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-html5/>
- Axarnet. (5 de Junio de 2018). *Diferencias entre servidores dedicados y VPS*. Recuperado el 9 de Marzo de 2019, de Axarnet: <https://www.axarnet.es/blog/servidores-dedicados-o-vps/>
- Bermúdez, L. T., & Rodríguez, L. F. (2013). *Investigación en la gestión empresarial*. Bogotá: ECOE EDICIONES.
- Caicedo, J. I. (4 de Mayo de 2015). *Web: ¿Qué es el Framework Bootstrap? Ventajas y Desventajas*. Recuperado el 16 de octubre de 2018, de Apuntes de Programación:
<http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>
- cenicafe. (s.f.). *Cenicafe*. Recuperado el 5 de Marzo de 2017, de Beneficio del Café II: Secado del Café Pergamino:
http://www.cenicafe.org/es/publications/cartilla_21._Secado_del_cafe.pdf
- ColCoffeeInsights. (Junio de 2012). *Detrás del café de Colombia: Perspectiva desde el origen*. Obtenido de Secado del café, otra de las claves para que el Café de Colombia sea de alta calidad: http://www.cafedecolombia.com/ccf-fnc-es/index.php/comments/el_secado_del_cafe_otra_de_las_claves_para_que_el_cafe_de_colombia_adquiera/

- Del valle Hernandez, L. (s.f.). *Escoger el mejor sensor de temperatura para Arduino*. Recuperado el 11 de Septiembre de 2018, de Programar Facil: <https://programarfacil.com/podcast/82-escoger-mejor-sensor-temperatura-arduino/>
- Delgado, H. (7 de Febrero de 2019). *Qué es JavaScript - Origen, evolución y características*. Obtenido de Akus.net Diseño Web: <https://disenowebakus.net/javascript.php>
- Dir.ca.gov. (2012). *Calor Excesivo en el Lugar de Trabajo: ¿Cómo prevenir enfermedades causadas por el calor en espacios de trabajo cubiertos?* Obtenido de <https://www.dir.ca.gov/chswc/WOSHTEP/SpecialistCourseMaterials/WOSHTEPIndoorHeatPreventionMaterialsParticipantsHandoutsSPANFINAL.pdf>
- DIYMarkers. (28 de Diciembre de 2013). *Mover motores paso a paso con arduino*. Recuperado el 12 de Septiembre de 2018, de DIYMarkers: <http://diymakers.es/mover-motores-paso-paso-con-arduino/>
- Echeverry Ocampo, D. A. (2010). Obtenido de Diseño y Simulacion de la Automatizacion de un secador Electromecanico de Cafe: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2987/66393E18.pdf?sequence=1>
- El Profe Garcia. (11 de octubre de 2017). *Como Automatizar Timbre de Colegio, Riego de Cultivos, Comedero de Animales*. Obtenido de YouTube: <https://www.youtube.com/watch?v=j9WSRhnVQc4>
- Energia controlada. (s.f.). *¿Qué es un Motorreductor?* Recuperado el 12 de Septiembre de 2018, de Energia Controlada: <https://www.energiacontrolada.com/faq/Que-es-un-motorreductor>
- Esqueda Elizondo, J. J. (Noviembre de 2002). *Matlab e Interfaces Gráficas*. Recuperado el 9 de Marzo de 2018, de Unicauca.edu.co: ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Identificacion/matlab_seminar/docs/Matlab6xConatec.pdf
- GONZALEZ, C., SANZ, J., & Oliveros, C. E. . (2014). *Control de caudal y temperatura de aire en el secado mecánico de café*. Obtenido de <http://biblioteca.cenicafe.org/handle/10778/503>
- Hetpro. (25 de Noviembre de 2017). *LM35 – El sensor de temperatura más popular*. Obtenido de Hetpro: <https://hetpro-store.com/TUTORIALES/lm35/>
- Hostingsaurio. (28 de Diciembre de 2017). *Qué es un servidor web*. Recuperado el 9 de Marzo de 2019, de Hostingsaurio.com: <https://hostingsaurio.com/que-es-un-servidor-web/>
- Ingesec. (s.f.). *Ingesec. Tecnología al Servicio de la Caficultura*. Obtenido de Equipos de Secado de Café: <http://www.ingeseclda.com/secado.htm>
- Iruela, J. (1 de Enero de 2016). *Los gestores de bases de datos más usados*. Obtenido de Revistadigital Inesem: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

jecrespom. (26 de Marzo de 2015). *Lenguaje de programación C++*. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>

Jesuites Educacion. (26 de Febrero de 2018). *Por qué elegir el gestor de base de datos MySQL*. Recuperado el 16 de octubre de 2018, de Jesuites Educacion Formacion Profesional: <https://fp.uoc.fje.edu/blog/por-que-elegir-el-gestor-de-base-de-datos-mysql/>

Laborda, J. (1 de octubre de 2016). *Introducción al ESP8266 y NodeMCU*. Obtenido de GitHub: <https://github.com/jaimelaborda/Planta-Twittera/wiki/1.-Introducci%C3%B3n-al-ESP8266-y-NodeMCU>

Landin, P. (12 de Febrero de 2018). *Tornillo sin fin: descripción y aplicaciones*. Obtenido de Pelandintecno - Tecnología Eso: <http://pelandintecno.blogspot.com/2018/02/tornillo-sin-fin-descripcion-y.html>

Lenguajes de programacion. (s.f.). *C# (C Sharp)*. Obtenido de Lenguajes de programacion.net: <https://lenguajesdeprogramacion.net/c-sharp/>

Leon Leon, A. (2012). *Academia*. Obtenido de Control de Temperatura para Secadora de Cafe: http://www.academia.edu/17048434/CONTROL_DE_TEMPERATURA_PARA_SECADORA_DE_CAFE

Microsoft. (19 de Julio de 2015). *Introducción al lenguaje C# y .NET Framework*. Obtenido de Microsoft: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

Microsoft Visual Studio .net. (s.f.). *Características de Visual Studio .NET*. Obtenido de Microsoft: <https://www.danysoft.com/free/vstudio.pdf>

Morales, R. (1 de Septiembre de 2014). *Lenguajes de programación: ¿qué son y para qué sirven?*. Obtenido de Colombia Digital: <https://colombiadigital.net/actualidad/articulos-informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html>

Mozilla. (13 de Septiembre de 2018). *¿Que es JavaScript?* Recuperado el 9 de Marzo de 2018, de MDN Web docs: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript

Multimedia. (26 de Marzo de 2013). *Lenguaje de Programacion C++*. Obtenido de Lenguaje de programacion 21: <http://lenguajedeprogramacion21.blogspot.com/>

Mussa, Y. (26 de Julio de 2016). *MySQL: uno de los gestores de base de datos más utilizado*. Recuperado el 16 de octubre de 2018, de FX2: <http://fx2.com.uy/mysql-un-aliado-para-la-gestion-de-base-de-datos>

Nadeau, C. (19 de Julio de 2018). *Configuración del nivel de seguridad de contraseñas (Professional y Enterprise)*. Recuperado el 21 de Septiembre de 2018, de Zendesk:

<https://support.zendesk.com/hc/es/articles/203663736-Configuraci%C3%B3n-del-nivel-de-seguridad-de-contrase%C3%B1as-Professional-y-Enterprise->

Omniblug. (24 de Julio de 2013). *Sensor de temperatura y humedad DHT11 - DHT22*. Obtenido de Omniblug: <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>

PHP.net. (s.f.). *Foreach*. Recuperado el 15 de Marzo de 2019, de PHP.net: <http://php.net/manual/es/control-structures.foreach.php>

Ramirez, P. (14 de Febrero de 2017). *Lenguaje de programación PHP: ¿Qué es y cuáles son sus características principales?* Obtenido de fireosoft: <https://fireosoft.com.co/blogs/lenguaje-de-programacion-php/>

Ramos Fuentes, M. (Diciembre de 2017). Efectos en Salud de la exposicion a altas temperaturas por desempeño laboral a la interperie. *Efectos en Salud de la exposicion a altas temperaturas por desempeño laboral a la interperie*. Chile: Institución de salud publica Ministerio de salud. Obtenido de www.ispch.cl/sites/default/files/NotaTecnicaCalor.pdf

Red Grafica Latinoamerica. (12 de octubre de 2018). *El lenguaje de programación PHP*. Obtenido de redgrafica: <http://redgrafica.com/El-lenguaje-de-programacion-PHP>

Remón, B. (23 de Septiembre de 2011). *Efectos sobre la salud en trabajos con altas temperaturas*. Recuperado el 12 de Septiembre de 2018, de Cen7dias: <http://www.cen7dias.es/contenido.php?bol=32&id=974&sec=4>

robotsia. (12 de octubre de 2015). *Robotsia*. Obtenido de ¿Qué es el Internet de las Cosas? Algunos ejemplos prácticos: <https://robotsia.com/2015/10/12/internet-de-las-cosas-iot-que-es-ejemplos-practicos/>

sistemasumma. (20 de Mayo de 2011). *Diccionario de datos*. Obtenido de Sistemas Umma: <https://sistemasumma.com/2011/05/20/diccionario-de-datos/>

Sosa, E. T. (2016). *Investogación Educativa: Fundamentos para la investigación formativa*. Recuperado el 13 de Abril de 2019, de eumed.net: <http://www.eumed.net/libros-gratis/2016/1553/1553.zip>

Tecnologia-informatica. (12 de Octubre de 2018). *Que es un lenguaje de programación*. Obtenido de Tecnologia- Informatica: <https://tecnologia-informatica.com/lenguaje-de-programacion/>

Tecnologias. (18 de Diciembre de 2013). *Raspberry PI*. Recuperado el 11 de Septiembre de 2018, de Blog Historia de la informatica: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

UNAD. (09 de Junio de 2011). *La Investigacion en la Escuela de Ciencias Básicas, Tecnología e Ingeniería*. Obtenido de https://amazonia.unad.edu.co/images/stories/ECBTI/LINEAS_DE_INVESTIGACION_ECBTI_2011_I_EXPLICADAS.pdf

- Urrego, J. F. (3 de marzo de 2019). *Crea sistemas POS - Inventarios y ventas con PHP7 y AdminLTE*. Obtenido de Udemy: <https://www.udemy.com/crea-sistemas-pos-inventarios-y-ventas-con-php7-y-adminlte/?ranMID=39197&ranEAID=XHBMnRHJeXE&ranSiteID=XHBMnRHJeXE-QASE0UcoPTTo4r3Lm.Eug5Q&LSNPUBID=XHBMnRHJeXE>
- Vistronica. (s.f.). *Motor Paso a Paso 5.6Kgcm Nema 17 Para Impresora 3D*. Recuperado el 13 de octubre de 2018, de Vistronica: <https://www.vistronica.com/robotica/motores/motor-paso-a-paso/motor-paso-a-paso-5-6kgcm-nema-17-para-impresora-3d-detail.html>
- Vistronica. (s.f.). *Motorreductor con caja reductora 6V 0.5kgCm 220rpm 1:48*. Recuperado el 13 de octubre de 2018, de Vistronica: <https://www.vistronica.com/robotica/motores/motorreductores/motorreductor-con-caja-reductora-6v-0-5kgcm-220rpm-1-48-detail.html>
- Woratek. (3 de Enero de 2012). *Características Raspberry Pi, el computador del tamaño de un llavero*. Obtenido de Woratek: <http://www.woratek.com/2012/01/03/caracteristicas-raspberry-pi/>

Anexo 1 - Encuesta

Nombres: _____

Apellidos: _____

Tiene Finca Propia: SI ☐ NO ☐

¿Cuál es el número de hectáreas que compone su finca?

- a) Entre 1 y 2 hectáreas
- b) Entre 3 y 5 hectáreas
- c) Entre 5 y 10 hectáreas
- d) Más de 10 hectáreas

¿Usted como caficultor Vende el café Mojado o Seco?

- a) Mojado
- b) Seco

En caso de venderse Mojado ¿Porque No realiza el proceso para secar el café?

- a) Situación Económica
- b) Falta de Tiempo
- c) Calidad obtenida del Café
- d) Otro ¿Cuál? _____

¿Conoce las diferentes características que debe cumplir el café en seca para su respectiva comercialización?

- a) Si
- b) No

¿Conoce el tiempo suficiente para determinar el punto exacto del café en seco?

- a) Si
- b) No

¿Qué tipo de Sistema utiliza para realizar el proceso de secado del café?

- a) Secador Solar
- b) Silo
- c) Otro ¿Cuál? _____

En su sistema de secado. ¿Cuánto tiempo demora en terminar el proceso de secado del café para su comercialización?

- a) 2 días
- b) 3 días
- c) 4 días

d) 5 días

¿Cada cuánto tiempo revuelve el café para el proceso de secado?

a) 30 Minutos

b) 60 Minutos (1 hora)

c) 90 Minutos (1 hora y media)

d) 120 Minutos (2 horas)

e) Otro ¿Cuál? _____

Usted vende su producción cafetera a:

a) Compras de Café del común

b) Minoristas

c) Compra de Café del Comité de Cafeteros del Municipio

d) Mayoristas

¿Usted Posee conexión a Internet?

a) Si

b) No

¿Usted tiene algunos de los siguientes implementos en su finca? Seleccione varios.

☐ Computador

☐ Tablet

☐ Celular Android

☐ Celular “Flecha”

¿Conoce alguna aplicación o sistema que le ayude a controlar la seca del café?

a) Si

b) No

¿Le gustaría que se implementara un sistema para controlar la seca del café?

a) Si

b) No

Gracias por el Tiempo que dedico para Responder esta Encuesta...

Anexo 2 – Resultados Encuesta

Link encuestas realizadas:

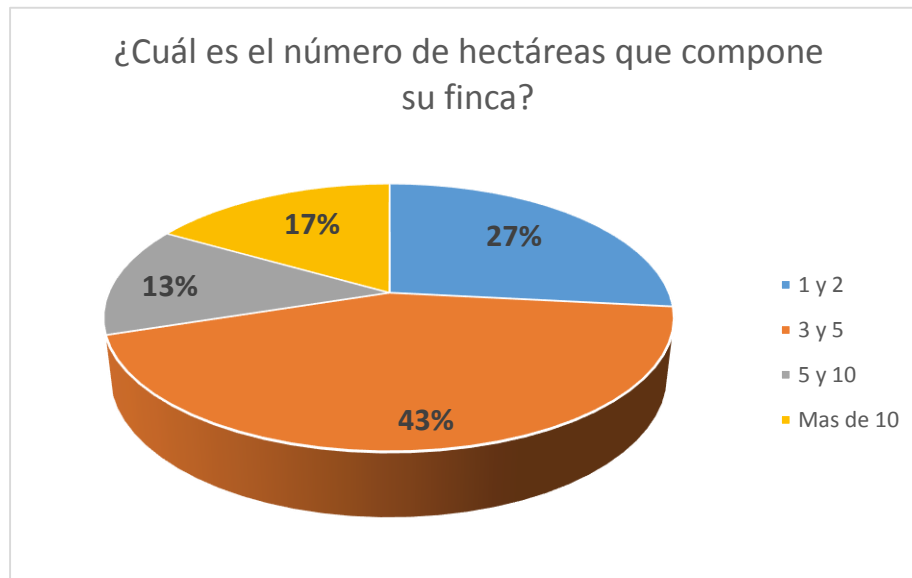
<https://drive.google.com/drive/folders/1pXicXtBNuX0YZKPCsUdoN-rNVGNkqOrZ?usp=sharing>

Gráficos de resultados

N° Hectáreas	¿Cuál es el número de hectáreas que compone su finca?
1 y 2	8
3 y 5	13
5 y 10	4
Más de 10	5
Total general	30

Autor: Hector Fabio Duran Montes

Figura 97 Resultado Encuesta - Pregunta 1



Fuente: El Autor

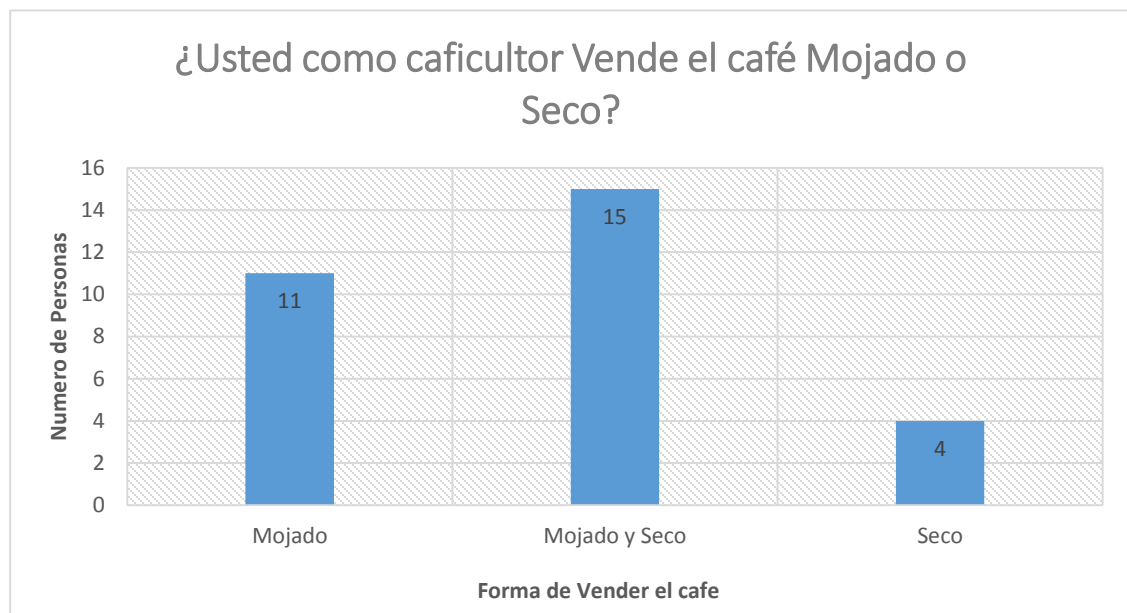
Análisis: Se Identifica que del 100% de las personas encuestadas el 43% tienen fincas entre 3 y 5 hectáreas y el 27% de los encuestados tienen fincas comprendidas entre 1 y 2

hectáreas el cual son los pequeños caficultores. Esto haciendo un total del 70% de la población ubicándolos en el factor fundamental a implementar el prototipo.

Forma de vender el producto	¿Usted como caficultor Vende el café Mojado o Seco?
Mojado	11
Mojado y Seco	15
Seco	4
Total general	30

Autor: Hector F. Duran

Figura 98 Resultado Encuesta - Pregunta 2



Fuente: El Autor

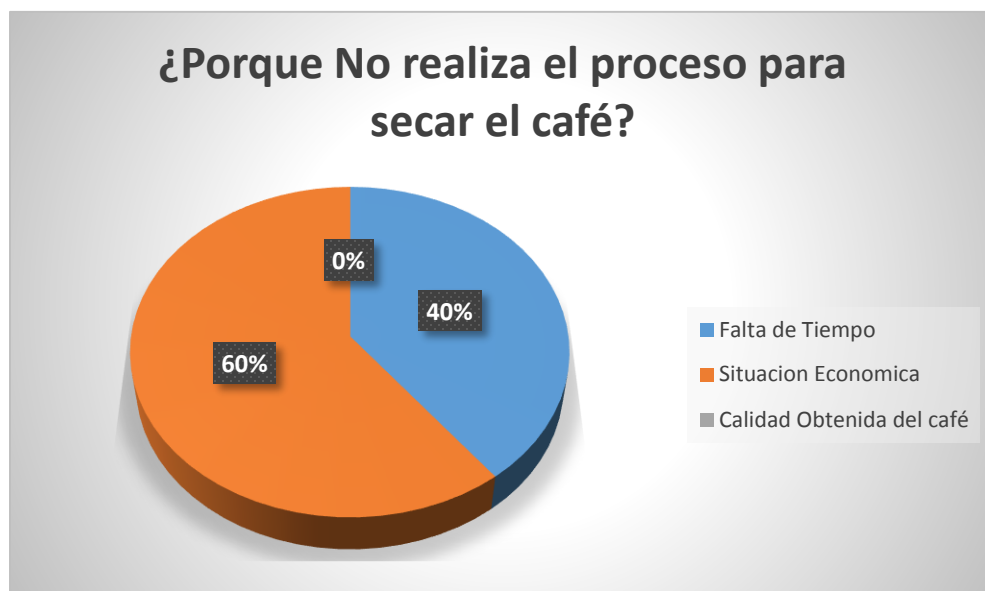
Análisis: Como factor primordial se identifica que tan solo 4 personas venden el producto del café seco convirtiéndolos en el 13% de la población total. El 37% pertenece a caficultores que venden el café mojado (11 personas) y un 50% de las personas que realizan la venta del producto de café mojado y seco. Se concluye que el 60% de la población vende el producto del café mojado.

¿Porque No realiza el proceso para secar el café?

Falta de Tiempo	17
Situación Económica	26
Calidad Obtenida del café	0
Total general	43

Autor: Hector Fabio Duran Montes

Figura 99 Resultado Encuesta - Pregunta 3



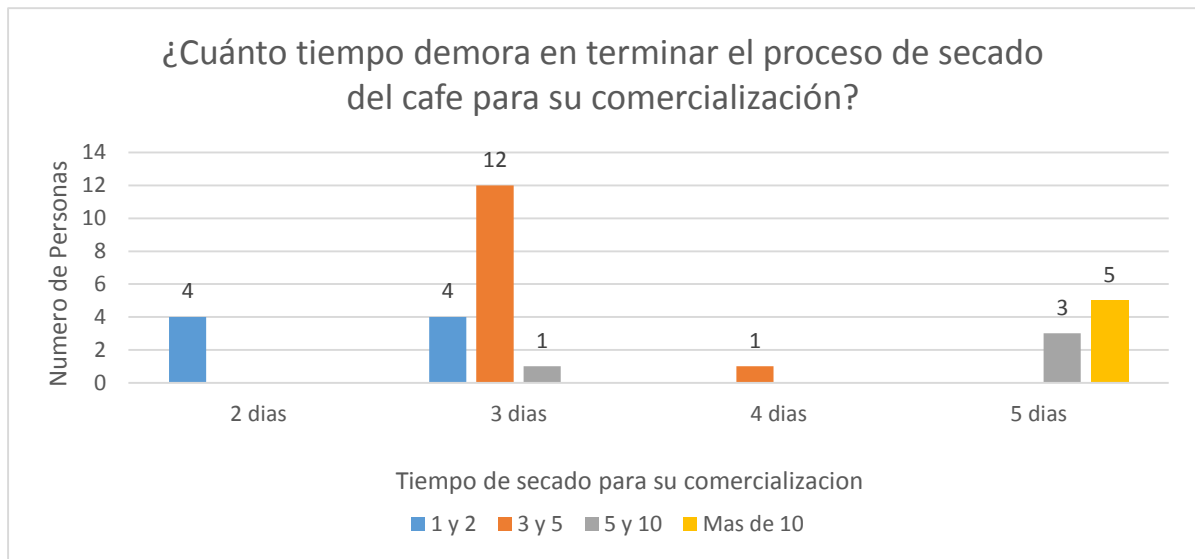
Fuente: El Autor

Análisis: Se puede identificar que el factor predominante es la “situación económica” con un 60% sobre el porqué de la No realización del proceso de secado del producto de café.

N° de Hectáreas	Tiempo de secado para su comercialización				Total general
	2 días	3 días	4 días	5 días	
1 y 2	4	4			8
3 y 5		12	1		13
5 y 10		1		3	4
Más de 10				5	5
Total general	4	17	1	8	30

Autor: Hector Fabio Duran Montes

Figura 100 Resultado Encuesta - Pregunta 4



Fuente: El Autor

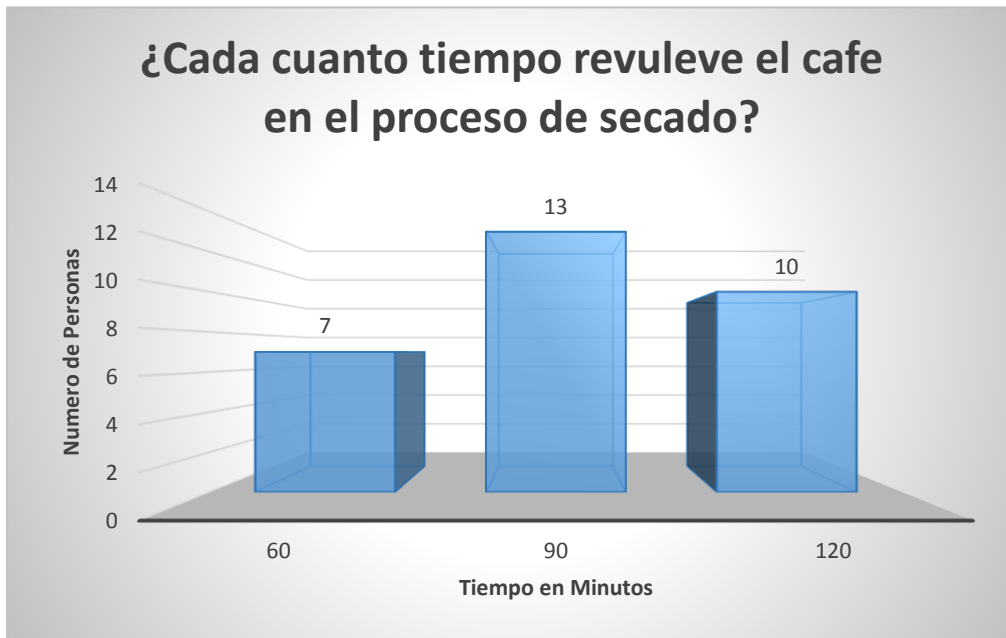
Análisis: Según la información se analiza que entre mayor número de hectáreas, mayor es el tiempo (en días) para secar el café y poder ser comercializado. Teniendo como análisis que solo 8 personas se demoran 5 días para secar el café y comercializarlo.

De las 22 personas restantes se interpretó que el 99% de los encuestados tienen un tiempo entre 2 y 3 días para realizar el proceso de secado del café y de igual manera ser comercializado.

Tiempo para revolver el café (minutos)	N° de Personas
60	7
90	13
120	10
Total general	30

Autor: Hector Fabio Duran Montes

Figura 101 Resultado Encuesta - Pregunta 5



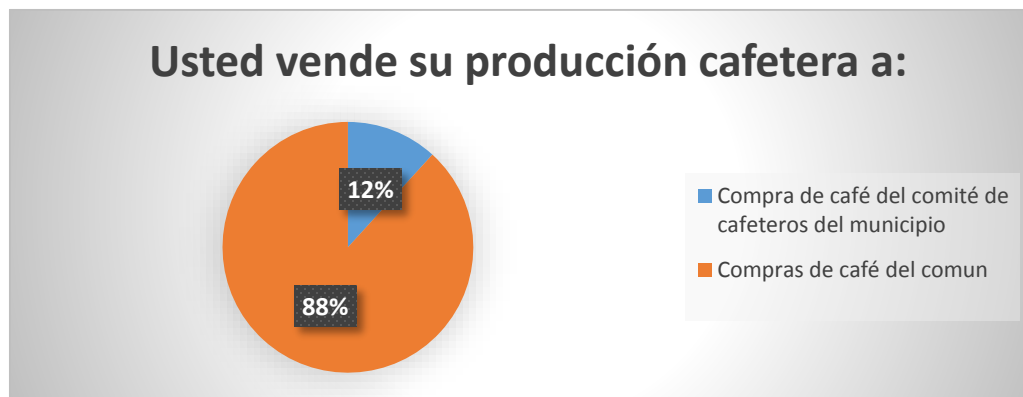
Fuente: El Autor

Análisis: en la gráfica se visualiza que el 77% de las personas encuestadas (13 y 10) realizan el proceso de revolver el café entre 90 minutos (1 hora y media) y 120 minutos (2 horas).

Usted vende su producción cafetera a:	
Compra de café del comité de cafeteros del municipio	4
Compras de café del común	30
Total general	34

Autor: Hector Fabio Duran Montes

Figura 102 Resultado Encuesta - Pregunta 6



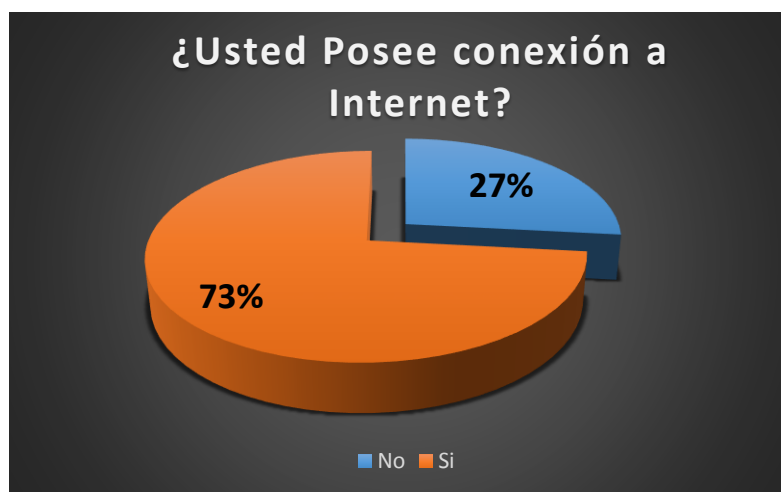
Fuente: El Autor

Análisis: las compras de café del común son el factor predominante en este ítem con un 88% sobre el resultado de las personas encuestadas. Esto quiere decir que el procedimiento que realizan los caficultores para el secado del café es inadecuado ya que en el comité de cafeteros es más exigente el producto del café para su respectiva compra y comercialización.

¿Usted Posee conexión a Internet?	
No	8
Si	22
Total general	30

Autor: Hector Fabio Duran Montes

Figura 103 Resultado Encuesta - Pregunta 7



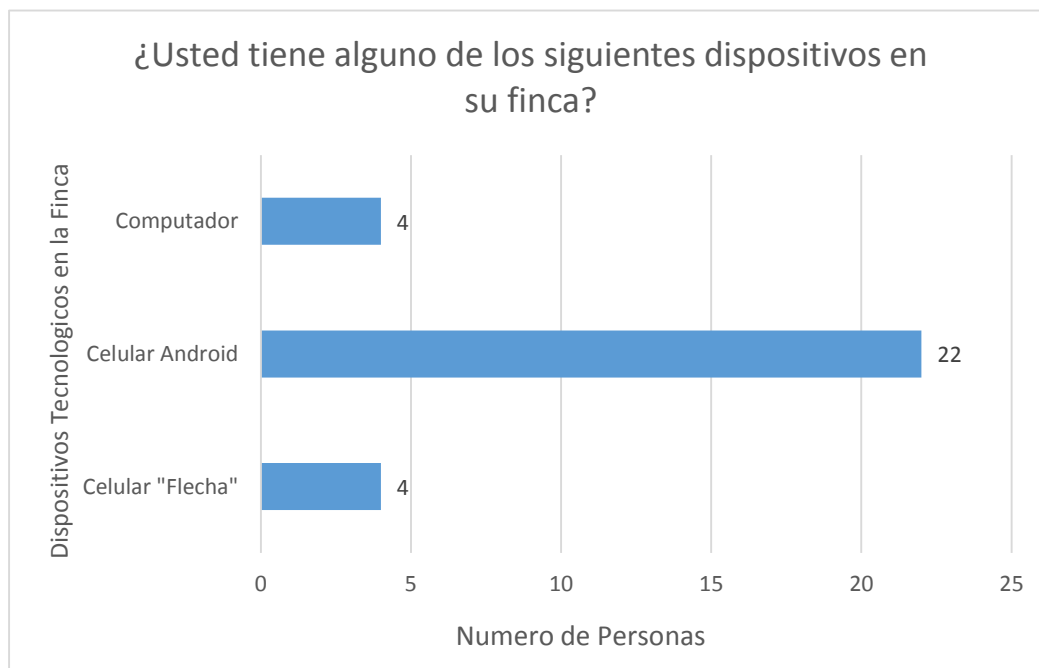
Fuente: El Autor

Análisis: De los 30 encuestados se identifica que el 73% poseen conexión a internet contando con un número de 22 caficultores.

¿Usted tiene alguno de los siguientes dispositivos en su finca?	N° de Personas
Celular "Flecha"	4
Celular Android	22
Computador	4
Total general	30

Autor: Hector Fabio Duran Montes

Figura 104 Resultado Encuesta - Pregunta 8



Fuente: El Autor

Análisis: El ítem “Celular Android” es el dispositivo que más poseen los caficultores con un número de 22 personas el cual hacen uso de este aparato electrónico.